



Departamento de Informática

Mestrado em Engenharia Informática

**Lofoscopia Assistida por Computador:  
Técnicas Computorizadas para  
Comparação de Impressões Digitais**

Nuno Chaves

Dissertação de Mestrado

Orientador: Prof. Dr. Luís Rato

Évora  
Dezembro 2010

Universidade de Évora  
Departamento de Informática

Nuno Chaves

**Lofoscopia Assistida por Computador: Técnicas  
Computorizadas para Comparação de Impressões Digitais**

*Dissertação apresentada à Universidade de Évora, como  
parte dos requisitos para a obtenção do título de Mestre  
em Engenharia Informática.*

Orientador: *Prof. Dr. Luís Rato*



485 660

Évora  
Dezembro 2010



*Aos meus pais, familiares e amigos.*



*Só aqueles que têm paciência  
para fazer coisas simples com perfeição  
é que irão adquirir habilidade  
para fazer coisas difíceis com facilidade.*

JOHANN CHRISTOPH VON SCHILLER (1759-1805)



# Agradecimentos

Sendo este um marco tão importante na minha vida, não poderia deixar passar sem agradecer a todos aqueles que o tornaram possível. Assim, sem qualquer ordem especial:

- Aos meus pais, por me terem dado a possibilidade de prosseguir os meus estudos, e pelo apoio que sempre me demonstraram.
- Ao professor João Barata e à professora Maria Helena Barata, pela ajuda na correcção ortográfica e da pontuação desta dissertação.
- Ao meu orientador, professor Luís Rato, pela sua orientação e disponibilidade para responder às minhas inúmeras dúvidas e questões.
- Ao chefe Manuel Marques, chefe da divisão de investigação criminal da polícia de segurança pública de Castelo Branco, pelo tempo que me disponibilizou para responder a dúvidas relacionadas com a lofoscopia e dactiloscopia.
- A todos os meus familiares e amigos, e todos os outros que directa ou indirectamente contribuíram para o meu sucesso académico.
- A ti Ana, por todo o apoio e dedicação que sempre me deste e continuas a dar.

Para todos vós, os meus mais sinceros agradecimentos.





## **Resumo**

Este trabalho consiste num estudo sobre técnicas para comparar impressões digitais. O estudo foi feito através do desenvolvimento de uma aplicação, que implementa as técnicas e algoritmos necessários, e da parametrização destes algoritmos, de forma a obter os melhores resultados possíveis. Foi realizada uma pesquisa bibliográfica sobre o problema e as principais técnicas utilizadas, sendo proposta e testada uma implementação de um sistema AFIS (Sistema Automatizado de Identificação de Impressões Digitais). Além dos métodos para fazer o processamento das imagens de impressões digitais, foi ainda desenvolvido um algoritmo, invariante à rotação e ao deslocamento, para comparação das mesmas. É ainda utilizado um módulo de classificação das impressões digitais, de modo a reduzir os tempos de pesquisa em bases de dados. Os testes realizados sobre a aplicação, revelaram tempos e resultados bastante aceitáveis, quando comparados com outras aplicações do mesmo tipo. O sistema proposto, foi desenvolvido, recorrendo a aplicações, cujo código fonte, foi disponibilizado como *software* livre.

**Palavras-chave:** Lofoscopia, Dactiloscopia, Impressão Digital, Processamento de Imagem, Reconhecimento de padrões.



## **Abstract**

This paper is a study on the techniques needed to match fingerprints. The study was made by developing an application that implements the necessary techniques and algorithms, and their parameters, in order to obtain the best results. A literature search was made on the problem, and through a combination of the most effective techniques usually used, is proposed and tested, an implementation of an AFIS system (Automatic Fingerprint Recognition System). Besides the methods needed to do the processing of fingerprint images, was also developed an invariant to rotation and displacement algorithm, to perform the fingerprint matching. It also uses a module for the classification of fingerprints in order to reduce the search time in databases. The results of the tests made on the application were very acceptable, when compared with other applications of the same kind. The proposed system was developed, using applications, whose source code was released as open-source.

**Keywords:** Lofoscopy, Dactyloscopy, Fingerprints, Image Processing, Pattern Matching.



# Lista de Abreviaturas

<b>AFIS</b>	Sistema Automatizado de Identificação de Impressões Digitais ( <i>Automatic Fingertip Identification System</i> ).
<b>FVC</b>	Competição de Verificação de Impressões Digitais ( <i>Fingerprint Verification Competition</i> ).
<b>EER</b>	Igual Taxa de Erro ( <i>Equal Error Rate</i> ).
<b>NIST</b>	Instituto Nacional de Normas e Tecnologia ( <i>National Institute of Standards and Technology</i> ).
<b>NBIS</b>	Software de Imagem Biométrico do NIST ( <i>NIST Biometric Image Software</i> ).
<b>FBI</b>	Gabinete de Investigação Federal ( <i>Federal Bureau of Investigation</i> ).
<b>DHS</b>	Departamento de Segurança da Pátria ( <i>Department Of Homeland Security</i> ).
<b>USA</b>	Estados Unidos da América ( <i>United States Of America</i> ).
<b>FVS</b>	Sistema de Comparação de Impressões Digitais ( <i>Fingerprint Verification System</i> ).
<b>SDK</b>	Pacote de Desenvolvimento de Software ( <i>Software Development Kit</i> ).
<b>DPI</b>	Pontos por polegada ( <i>Dots Per Inch</i> ).
<b>PCASYS</b>	Módulo de classificação da aplicação NBIS ( <i>NBIS Classifier</i> ).

---

<b>NFIQ</b>	Algoritmo de Melhoria de Imagem do NBIS ( <i>NIST Fingertip Image Quality</i> ).
<b>MINDTCT</b>	Módulo de detecção de minúcias do NBIS ( <i>NBIS Minutia Detector</i> ).
<b>BOZORTH3</b>	Algoritmo de comparação de Impressões Digitais do NBIS ( <i>NBIS Fingerprint Matching Algorithm</i> ).
<b>FFT</b>	Transformada Rápida de Fourier ( <i>Fast Fourier Transform</i> ).
<b>CN</b>	Número de Cruzamentos ( <i>Crossing Number</i> ).

# Lista de Figuras

2.1	Exemplo de duas impressões digitais.	6
2.2	Exemplo de vales e cristas papilares.	7
2.3	Posição do núcleo em duas impressões digitais.	7
2.4	Posição do delta em duas impressões digitais.	8
2.5	Detalhes de Galton com respectiva legenda.	9
2.6	Impressões digitais que apresentam dificuldades para o seu reconhecimento.	11
3.1	Resultado da aplicação dos filtros numa impressão digital nova.	16
3.2	Resultado da comparação de duas impressões digitais.	17
3.3	Inserção de uma imagem na base de dados utilizando o Verifinger.	19
3.4	Alguns exemplos de impressões digitais utilizadas na competição.	20
4.1	Classes do sistema de classificação de Henry.	22
4.2	Mascara 9x9 para o cálculo do mapa direccional.	24
4.3	Exemplo do método de suavização.	27
4.4	Mapa direccional da impressão digital.	27
4.5	Exemplo de cálculo do índice de Poincaré.	28
4.6	Cálculo matemático do índice de Poincaré.	29
4.7	Classificação utilizando o índice de Poincaré.	29
5.1	Diagrama do sistema.	31
5.2	Impressão digital e seu histograma antes da normalização do contraste.	33
5.3	Impressão digital e seu histograma após a normalização do contraste.	33
5.4	Impressão digital antes e depois do filtro de mediana.	34
5.5	Blocos de uma impressão digital antes e depois da aplicação do filtro.	36
5.6	Impressão digital após a aplicação do <i>Root Filtering</i> .	36
5.7	Fragmento de impressão digital e sinusóide com a mesma direcção.	37
5.8	Diferenças de espaçamento entre as cristas, ao longo da imagem.	38
5.9	Janela orientada para o cálculo do vector <i>x-signature</i>	38
5.10	Impressão digital otimizada pelo filtro de <i>Gabor</i> .	40
5.11	Binarização de uma impressão digital com limiar de 127.	41
5.12	Impressão digital binarizada utilizando binarização com limiar fixo e binarização adaptativa.	41
5.13	Vizinhança de 8 pixeis.	42
5.14	Exemplo do cálculo do número de conectividade ( $C(p) = 1$ ).	43



---

5.15	Primeira condição do algoritmo de esqueletização.	44
5.16	Segunda condição do algoritmo de esqueletização.	44
5.17	Terceira e quarta condições do algoritmo de esqueletização.	44
5.18	Condições do segundo passo do algoritmo de esqueletização.	45
5.19	Esqueletização de uma impressão digital.	45
5.20	Efeito escada.	46
5.21	Máscaras para remoção de pontos isolados, h-conectados e com espessura maior do que 1.	46
5.22	Máscara para remoção do efeito de escada.	46
5.23	Localização do pixel central e seus vizinhos.	47
5.24	Máscara para remoção dos pixels que contornam o objecto.	47
5.25	Impressão digital e respectivo mapa direccional.	48
5.26	Representação gráfica do filtro complexo para a extracção do ponto de referência.	48
5.27	Resposta do filtro complexo aplicado a uma impressão digital.	49
5.28	Localização do ponto de referência, numa impressão digital.	51
5.29	Vizinhança de 8 pixels, para o cálculo do número de cruzamentos.	51
5.30	Propriedades do algoritmo número de cruzamentos.	52
5.31	Exemplo de padrões no cálculo do número de cruzamentos.	52
5.32	Estrutura das minúcias extraídas pelo algoritmo CN.	52
5.33	Exemplos de falsas minúcias.	53
5.34	Validação de uma minúcia do tipo terminação.	54
5.35	Rejeição de uma minúcia do tipo terminação.	55
5.36	Validação de uma minúcia do tipo bifurcação.	55
5.37	Rejeição de uma minúcia do tipo bifurcação.	55
5.38	Minúcias extraídas pelo algoritmo CN, antes da validação.	56
5.39	Minúcias consideradas válidas, após o algoritmo de validação.	56
5.40	Ponto de referência como origem de um sistema de eixos para o cálculo do ângulo das minúcias.	57
5.41	Representação de uma impressão digital, na forma de uma lista.	58
6.1	Fluxograma do algoritmo de comparação.	60
6.2	Exemplo de uma representação de uma impressão digital.	64
6.3	Base de dados com 3 impressões digitais pertencentes à mesma classe.	65
6.4	Minúcias com a mesma distância da impressão digital a comparar.	65
6.5	Minúcias de $FBD_2$ em comum com $F_C$ .	66
6.6	Minúcias de $FBD_3$ em comum com $F_C$ .	66
6.7	Lista resultante da aplicação do algoritmo de comparação de impressões digitais.	67
7.1	Diagrama da aplicação desenvolvida.	69
7.2	Arquitectura da aplicação desenvolvida.	74
7.3	Fluxo de informação entre os módulos da aplicação.	75
7.4	Módulo de adição de imagens à base de dados.	75
7.5	Módulo de identificação de impressões digitais.	76
7.6	Relação entre o Root Filtering e o Filtro de Gabor.	78

---

7.7	Relação entre o total de minúcias extraídas, com e sem validação.	80
7.8	Representação da estrutura da base de dados.	81
7.9	Tempo de execução dos algoritmos.	83
7.10	Distribuição da identificação das imagens originais.	85
7.11	Distribuição dos resultados da identificação, usando imagens danificadas com ruído.	86
7.12	Distribuição dos resultados do teste com imagens deslocadas.	87
7.13	Distribuição dos resultados da identificação com impressões que apresentam rotações.	88
7.14	Distribuição dos resultados do teste com impressões digitais parciais.	89
7.15	Distribuição dos resultados da identificação das imagens mal classificadas.	91



# Lista de Tabelas

7.1	Características do computador de desenvolvimento e testes.	70
7.2	Impressões digitais seleccionadas por base de dados.	71
7.3	Distribuição das impressões digitais seleccionadas, em classes.	72
7.4	Origem do código fonte dos diversos algoritmos.	73
7.5	Tempo de execução de cada algoritmo.	82
7.6	Resultados da identificação usando as imagens originais.	84
7.7	Resultados da identificação usando imagens danificadas com ruído.	85
7.8	Erros verificados na identificação de imagens com ruído.	86
7.9	Resultados da identificação de impressões deslocadas.	87
7.10	Erros verificados na identificação de impressões deslocadas.	87
7.11	Resultados da identificação de impressões digitais com rotações.	88
7.12	Erros verificados na identificação com impressões rotacionadas.	89
7.13	Resultados da identificação usando imagens de impressões parciais.	89
7.14	Erros verificados na identificação de impressões parciais.	90
7.15	Resultados da identificação usando as imagens mal classificadas.	90
7.16	Total de identificações correctas e incorrectas.	91
7.17	Total de erros verificados na identificação.	92



# Lista de Algoritmos

6.1	Pseudo-código para o cálculo das minúcias comuns entre duas representações.	61
6.2	Pseudo-código para a comparação das minúcias.	62
7.1	Código necessário para utilizar os métodos do ImageJ.	77
7.2	Utilização do filtro de esqueletização do ImageJ.	79



# Sumário

<b>Agradecimentos</b>	<b>x</b>
<b>Resumo</b>	<b>xii</b>
<b>Abstract</b>	<b>xiv</b>
<b>Lista de Abreviaturas</b>	<b>xvii</b>
<b>Lista de Figuras</b>	<b>xix</b>
<b>Lista de Tabelas</b>	<b>xxi</b>
<b>Lista de Algoritmos</b>	<b>xxiii</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Motivação	2
1.2 Objectivos e contribuições	2
1.3 Bases de dados de impressões digitais	3
1.4 Estrutura da dissertação	3
<b>2 Lofoscopia e Dactiloscopia</b>	<b>5</b>
2.1 Dactiloscopia	6
2.1.1 Pontos Singulares	7
2.1.2 Pontos Característicos	8
2.1.3 Dificuldades no reconhecimento de impressões digitais	9
<b>3 Estado da arte e aplicações</b>	<b>13</b>
3.1 Estado da Arte	13
3.2 Aplicações AFIS	16
3.2.1 eFinger	16
3.2.2 Fingerprint Verification System	17
3.2.3 NIST Biometric Image Software	18
3.2.4 Verifinger	19
3.2.5 Fingerprint Verification Competition	19



---

<b>4</b>	<b>Metodologias de Classificação</b>	<b>21</b>
4.1	Sistema de Classificação de Henry	21
4.2	Métodos de classificação	22
4.3	Técnicas de Classificação	24
4.3.1	Cálculo do mapa direccional	24
4.3.2	Suavização do mapa direccional	26
4.3.3	Classificação pelo índice de Poincaré	28
4.4	Origem do código utilizado	30
<b>5</b>	<b>Técnicas para o Processamento de Impressões Digitais</b>	<b>31</b>
5.1	Pré-processamento	32
5.1.1	Normalização do contraste	32
5.1.2	Filtro de Mediana	34
5.1.3	Melhoramento das cristas	34
5.1.3.1	Root Filtering	35
5.1.3.2	Filtros de Gabor	37
5.1.4	Binarização Adaptativa	40
5.1.5	Esqueletização	42
5.1.5.1	Limpeza do Esqueleto	46
5.2	Extracção de Características	47
5.2.1	Cálculo do ponto de Referência	47
5.2.2	Extracção de Minúcias	51
5.3	Pós-processamento	53
5.3.1	Validação de Minúcias	53
5.3.2	Representação da Impressão Digital	56
5.4	Origem do código utilizado	58
<b>6</b>	<b>Identificação com impressões digitais</b>	<b>59</b>
6.1	Algoritmo de comparação proposto	59
6.2	Exemplo de aplicação do algoritmo proposto	64
6.3	Origem do código utilizado	67
<b>7</b>	<b>Aplicação Desenvolvida: Fingers</b>	<b>69</b>
7.1	Descrição da base de dados	70
7.2	Plataforma de desenvolvimento e implementações	72
7.3	Arquitectura da aplicação	74
7.3.1	Ambiente Gráfico	75
7.3.2	Classificação	76
7.3.3	Normalização do Contraste e Filtro de Mediana	76
7.3.4	Root Filtering e Filtro de Gabor	77
7.3.5	Binarização	78
7.3.6	Esqueletização e respectiva limpeza	79
7.3.7	Crossing Number e Validação de minúcias	79
7.3.8	Cálculo do ponto de referência	80
7.3.9	Construção da representação	81

---

7.3.10	Comparação de impressões digitais	81
7.4	Complexidade temporal	82
7.5	Testes à aplicação	83
7.5.1	Impressões Digitais Originais	84
7.5.2	Impressões Digitais com Ruído	85
7.5.3	Impressões Digitais com Deslocamento	86
7.5.4	Impressões Digitais com Rotações	88
7.5.5	Impressões Digitais Parciais	89
7.5.6	Imagens Mal Classificadas	90
7.6	Considerações Finais	91
<b>8</b>	<b>Conclusões e trabalho futuro</b>	<b>93</b>
	<b>Glossário</b>	<b>95</b>
	<b>Referências Bibliográficas</b>	<b>98</b>
<b>A.</b>	<b>Máscaras para limpeza do esqueleto</b>	<b>105</b>
<b>B.</b>	<b>Diagrama do algoritmo de comparação de impressões digitais</b>	<b>107</b>
<b>C.</b>	<b>Diagrama do módulo de introdução de impressões digitais na base de dados</b>	<b>109</b>
<b>D.</b>	<b>Código fonte do algoritmo Root Filtering</b>	<b>111</b>
<b>E.</b>	<b>Exemplos de impressões digitais utilizadas nos testes</b>	<b>113</b>
	<b>Índice Remissivo</b>	<b>115</b>



# Capítulo 1

## Introdução

Há mais de um século atrás, Alphonse Bertillon<sup>1</sup> inventou e desenvolveu a primeira técnica conhecida para identificação pessoal (sistema antropométrico, mais conhecido como sistema de Bertillon), cujo método se baseava na utilização de medidas do corpo humano, de forma a resolver crimes.

No entanto, ainda o seu sistema estava a dar os primeiros passos, quando em 1893, os tribunais aceitaram que não havia dois indivíduos com a mesma impressão digital. Na presença desta aceitação, por parte dos tribunais, os departamentos de investigação criminal viram imediatamente grande potencial neste novo método de identificação pessoal. Havia então nascido a identificação de pessoas através de impressões digitais.

Apesar de todos os métodos improvisados para aumentar a eficiência da identificação manual de impressões digitais, estes continuam a ser bastante demorados. Além disso, a comparação de duas impressões digitais, provocava no especialista um grande desgaste físico e um aumento do tédio, devido ao tempo necessário para fazer o trabalho.

Devido a estes factores e à imensa procura dos serviços de identificação através de impressões digitais, as agências de investigação criminal iniciaram pesquisas que permitissem a captura de impressões digitais através de meios electrónicos, assim como o seu reconhecimento através de uma representação digital da mesma. Estas pesquisas levaram ao desenvolvimento de Sistemas Automatizados de Comparação de Impressões Digitais (AFIS), nas últimas décadas.

---

<sup>1</sup>Agente da polícia francesa e investigador biométrico, nascido em 1853

### 1.1 Motivação

A unicidade das impressões digitais permite a sua utilização para identificação de pessoas, uma vez que, não se conhecem duas pessoas que apresentem a mesma impressão digital. No entanto, os processos manuais de identificação das mesmas, são bastante limitados, o que torna a eficiência de todo o processo bastante reduzida. A maior limitação existente prende-se com o tamanho da impressão digital, uma vez que este é bastante reduzido, tornando difícil a identificação dos pontos característicos a olho nu. Por outro lado, as bases de dados de impressões digitais costumam ser de grandes dimensões, o que torna todo o processo manual, bastante demorado.

Uma aplicação clássica de identificação por impressões digitais, é a área criminal. No entanto, o mesmo estilo de sistema pode ser utilizado noutras áreas, sempre que seja necessário comprovar a identidade de uma pessoa. Áreas como sistemas de segurança e equipamentos de controle de acessos, de forma a autorizar ou negar o acesso de pessoas a áreas restritas, autenticação em sistemas informáticos, ou mesmo marcadores de "ponto" em empresas, são utilizações possíveis para sistemas de identificação.

Os métodos normais de identificação pessoal, usualmente utilizam outro tipo de informação, tal como uma senha ou mesmo uma chave ou um cartão. O problema é que este tipo de informação pode ser partilhado, esquecido, furtado ou perdido.

Por estes motivos, justifica-se o crescente desenvolvimento de sistemas automáticos, tendo como base, os avanços tecnológicos nas áreas de processamento de imagens e reconhecimento de padrões. A utilização destes sistemas, permite acelerar o reconhecimento de uma pessoa, com auxílio de sistemas computadorizados, sem a necessidade de um especialista na área.

### 1.2 Objectivos e contribuições

O objectivo deste trabalho é fazer um estudo sobre as técnicas necessárias para comparar impressões digitais. Esse estudo, engloba as técnicas necessárias para classificar as impressões digitais, os algoritmos que permitem melhorar as mesmas, terminando num algoritmo que compara as impressões digitais entre si.

Este estudo foi feito sobre a forma de uma aplicação informática, na qual foram implementadas todas as técnicas necessárias.

As principais contribuições desta dissertação são:

- Análise descritiva dos métodos e algoritmos utilizados.

### 1.3. BASES DE DADOS DE IMPRESSÕES DIGITAIS

---

- Implementação de um sistema de identificação de impressões digitais, baseado na localização de minúcias e pontos de referência.
- Desenvolvimento de um algoritmo de comparação de impressões digitais.

## 1.3 Bases de dados de impressões digitais

Para testar o desempenho dos métodos e algoritmo propostos, foi construída uma base de dados de impressões digitais, recorrendo a diversos conjuntos disponíveis para *download* na *internet*.

Foram utilizadas imagens do conjunto de impressões digitais da *Fingerprint Verification Competition* (referentes às edições de 2000, 2002 e 2004), impressões digitais capturadas com sensores ópticos *U.are.U 4000* e *Cross Match Verifier 300*, impressões digitais disponibilizadas pela revista *online* *Ciência Forense*, e ainda, o conjunto de impressões utilizadas no concurso *Singular Point Detection Competition 2010*.

As imagens utilizadas para testar a aplicação desenvolvida, foram construídas com base nas impressões digitais seleccionadas anteriormente. Para cada teste, foi seleccionado um subconjunto de 50 impressões digitais, às quais foram aplicadas as seguintes distorções, consoante o objectivo do teste:

- Adição de ruído.
- Deslocamento da impressão digital.
- Rotação da impressão digital.
- Remoção de partes da impressão digital.

## 1.4 Estrutura da dissertação

Esta dissertação está dividida em 8 capítulos distintos, sendo o capítulo 1, esta introdução.

No capítulo 2, é feita uma breve descrição da lofoscopia e dactiloscopia, assim como os principais problemas no processo de identificação de impressões digitais.

O capítulo 3, é composto pelo estado da arte, onde são analisadas algumas aplicações, que permitem fazer a comparação de impressões digitais.

No capítulo 4, é apresentado o problema da classificação, assim como os métodos e algoritmos utilizados, com o objectivo de classificar uma impressão digital.

## **CAPÍTULO 1. INTRODUÇÃO**

---

Os métodos e técnicas necessárias, para fazer o processamento das imagens de impressões digitais, são descritos em detalhe no capítulo 5.

O algoritmo de comparação de impressões digitais, desenvolvido neste trabalho, é apresentado em detalhe no capítulo 6.

No capítulo 7, é feita uma descrição da aplicação desenvolvida, juntamente com as ferramentas utilizadas no seu desenvolvimento, os testes realizados sobre a aplicação, e respectivos resultados.

Finalmente, o capítulo 8, que se refere à conclusão e trabalho futuro.

## Capítulo 2

# Lofoscopia e Dactiloscopia

A Lofoscopia é a ciência que estuda os desenhos formados pelas cristas papilares das impressões digitais, palmas das mãos e plantas ou palmas dos pés. O termo tem origem das palavras gregas, *lophos* que significa relevo e *skopein* que quer dizer exame [1].

Esta ciência assenta em três princípios [1, 2, 3], que lhe garantem fiabilidade:

- Perenidade

Os desenhos papilares que existem na ponta dos dedos, nas palmas das mãos e dos pés, não variam no seu mais pequeno detalhe, quer em relação ao número, forma, situação e direcção das suas cristas, a partir do 6º mês de gestação, ou seja, não sofrem alterações desde a sua criação até ao apodrecimento dos tecidos.

- Imutabilidade

Está directamente relacionada com a perenidade e significa que os desenhos não sofrem alterações fisiológicas, patológicas ou voluntárias, quer no número de cristas, formas ou direcções das mesmas. No caso de haver alguma destruição parcial dos desenhos, através de lesões menores, os mesmos regeneram-se sempre com o mesmo padrão. No caso de lesões de maior gravidade, podem aparecer cicatrizes permanentes, que não representam um obstáculo à identificação, podendo mesmo fornecer novas características identificativas.

- Variabilidade

Os desenhos papilares são todos diferentes uns dos outros. Eles diferem de dedo para dedo, de mão para mão, de pessoa para pessoa. Não existem duas impressões papilares iguais.

Estes três princípios permitem a identificação, registo e classificação destes desenhos, de



forma a serem utilizados para fins de identificação de um indivíduo, através da comparação dos mesmos [3].

A lofoscopia divide-se em três disciplinas diferentes, que estudam as várias zonas do corpo humano, que apresentam desenhos papilares. Elas são:

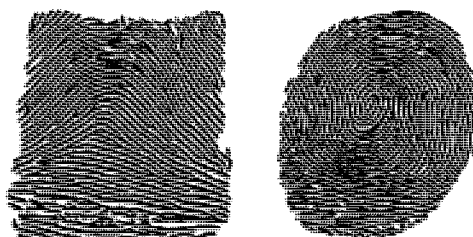
- Dactiloscopia  
Estuda o desenho das extremidades dos dedos.
- Quiroscopia  
Estuda os desenhos das palmas das mãos.
- Pelmatoscopia  
Estuda os desenhos das plantas dos pés.

Dado que a maioria dos estudos efectuados sobre a lofoscopia se centram usualmente na dactiloscopia, é usual utilizar as duas palavras como sinónimos.

### 2.1 Dactiloscopia

A palavra dactiloscopia tem origem nas palavras gregas, *dactilo* que significa dedo e *skopein* que significa exame. Assim, a dactiloscopia define-se como sendo o estudo detalhado dos desenhos que formam as linhas das papilas digitais, com a finalidade de identificar indivíduos [1].

Uma impressão digital é uma reprodução das linhas das papilas digitais num qualquer suporte. A sua característica estrutural mais evidente é um padrão composto por vales e cristas intercalados. Além desta evidência, uma impressão digital é ainda composta por pontos característicos (minúcias), e pontos singulares (deltas e núcleos) [4]. Na figura 2.1 apresenta-se um exemplo de duas impressões digitais.

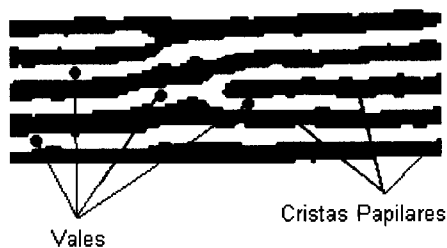


**Figura 2.1** Exemplo de duas impressões digitais.

## 2.1. DACTILOSCOPIA

---

As cristas papilares correspondem às linhas pretas, que se observam nas impressões digitais. Não têm sempre a mesma espessura, uma vez que a mesma varia de dedo para dedo, e de pessoa para pessoa. Os vales correspondem às linhas brancas, que acompanham intercaladamente as cristas papilares, reflectindo o seu comportamento [1]. Na figura 2.2, é possível ver um exemplo de vales e cristas papilares, numa impressão digital.



**Figura 2.2** Exemplo de vales e cristas papilares.

As características que constituem uma impressão digital são utilizadas de duas formas distintas. Por um lado, é possível olhar globalmente para o comportamento das cristas papilares de forma a classificar essa impressão digital num grupo. Por outro lado, é possível fazer corresponder a impressão digital a uma pessoa, através da análise das minúcias existentes na mesma.

### 2.1.1 Pontos Singulares

Os pontos singulares são as primeiras características observadas ao se realizar uma análise de uma impressão digital. O primeiro ponto singular que se destaca é o núcleo, uma vez que este corresponde ao ponto central de uma impressão digital, sendo envolvido por cristas papilares. Dependendo da forma como as cristas estão dispostas, ao longo do centro, é possível classificar a impressão digital em várias classes (ver capítulo 4). Existem impressões digitais nas quais não é possível verificar a existência de um núcleo. Nestes casos, o núcleo é definido no ponto onde as cristas papilares apresentam a maior curvatura [5]. Na figura 2.3, é possível ver um exemplo de um núcleo de uma impressão digital.



**Figura 2.3** Posição do núcleo em duas impressões digitais.

Outro ponto singular é o delta. Este ponto pode ser formado de duas maneiras distintas: através da bifurcação de uma crista ou através de uma convergência brusca de duas cristas paralelas. Este ponto singular tem um papel muito importante na classificação de impressões digitais, uma vez que a quantidade de deltas numa impressão influencia a sua classe [6]. A figura 2.4 apresenta dois exemplos de deltas em duas impressões digitais diferentes.



**Figura 2.4** Posição do delta em duas impressões digitais.

### 2.1.2 Pontos Característicos

Os pontos característicos (minúcias), também designados por detalhes de Galton, são os pontos resultantes das várias formas das cristas papilares [7]. Alguns destes pontos resultam da terminação ou bifurcação das cristas papilares, e definem a unicidade das impressões digitais.

As minúcias podem ser classificadas em duas categorias: aspectos básicos e aspectos compostos. Os aspectos básicos são cristas finais e cristas bifurcadas. Os aspectos compostos são ilhas, cristas curtas, esporas e cruzamentos [8]. Galton definiu este conjunto de aspectos [7], no entanto, como os aspectos compostos são construídos através dos aspectos básicos, é possível encontrar todos os aspectos, procurando apenas por estes últimos. Deste modo, as minúcias são resumidas a apenas cristas finais e cristas bifurcadas.

Na figura 2.5, é possível observar os diferentes tipos de padrões, identificados por Galton, onde A e B representam os aspectos básicos e C-F representam os aspectos compostos.

Em baixo, segue uma breve explicação de cada um dos aspectos:

- **Cristas finais**  
São resultantes da súbita terminação de uma crista papilar.
- **Cristas bifurcadas**  
Resultam da separação de uma crista papilar, em duas outras.

## 2.1. DACTILOSCOPIA

---

- Ilhas

São duas bifurcações ligadas entre si, ou seja, uma crista proveniente da direita que bifurca e se liga a outra crista proveniente da esquerda, que também bifurca.

- Cristas curtas

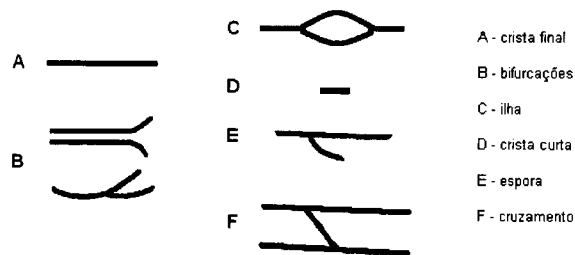
São iguais às cristas finais, mas de dimensões bastante reduzidas.

- Esporas

São formadas pela combinação de bifurcações com cristas finais.

- Cruzamentos

São definidos como sendo duas bifurcações que apresentam um caminho entre si próprias.



**Figura 2.5** Detalhes de Galton com respectiva legenda.

Para que duas impressões digitais sejam identificadas como iguais, é necessário que exista correspondência entre pelo menos 12 pontos característicos [9].

### 2.1.3 Dificuldades no reconhecimento de impressões digitais

O reconhecimento de impressões digitais, ainda apresenta, hoje em dia, diversas dificuldades, devido sobretudo às diferenças resultantes da recolha de impressões digitais de um mesmo dedo. Os principais factores [4] que contribuem para estes problemas são:

- Deslocamento

Na recolha da impressão digital, e sua posterior digitalização para utilização na identificação de um indivíduo, a área digitalizada muito provavelmente não corresponderá com a área armazenada na base de dados. Por exemplo, uma impressão digital que é digitalizada para comparação com uma outra igual armazenada na base de dados, pode estar impressa mais para a esquerda, ou mais para a direita, ou para cima, ou para baixo do que

## CAPÍTULO 2. LOFOSCOPIA E DACTILOSCOPIA

---

a sua correspondente. No caso de impressões digitais recolhidas com leitores biométricos<sup>1</sup>, este deslocamento é provocado pelo pressionar do dedo em diferentes localizações do sensor.

- Rotação

Muito dificilmente duas impressões digitais são recolhidas com o mesmo ângulo. Assim, é bastante provável que uma impressão que seja recolhida apresente diferenças em termos rotacionais em relação à mesma impressão que está guardada na base de dados.

- Sobreposição parcial

A existência de deslocamentos e de rotações provoca que possa existir apenas uma pequena área de sobreposição entre a imagem guardada na base de dados e a imagem que se utiliza para a comparação.

- Deformação não-linear

Sempre que se pressiona o dedo em alguma superfície, devido à elasticidade da pele, é provocada uma compressão e/ou dilatação da impressão digital. Assim, as diferentes pressões efectuadas na altura da recolha das impressões digitais fazem com que duas impressões do mesmo dedo possam parecer um pouco diferentes.

- Outro tipo de problemas, tais como cicatrizes e a oleosidade da pele no momento da captura da impressão digital; ruídos provocados por sujidade e poeira; erros durante a fase de extracção de características.

Na figura 2.6, é possível observar alguns dos tipos de dificuldades encontradas no reconhecimento de impressões digitais. Cada linha apresenta duas impressões digitais do mesmo dedo.

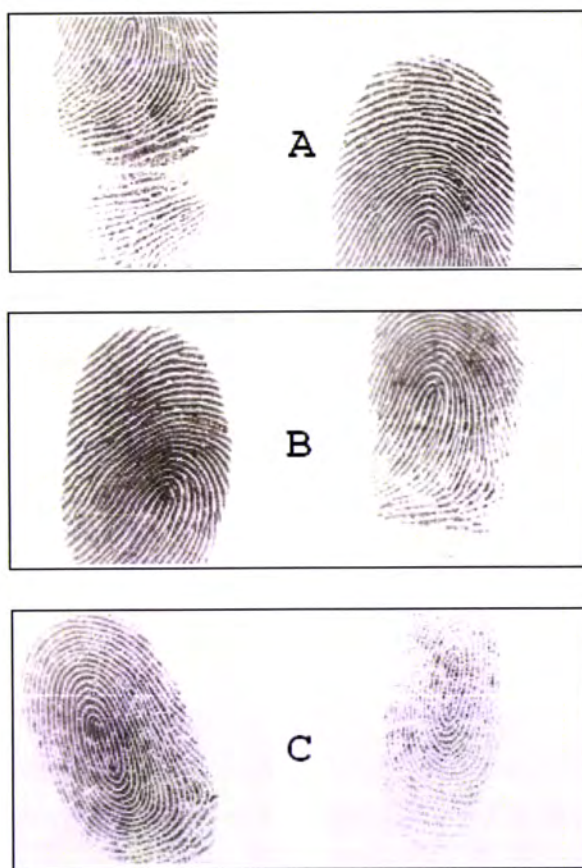
Em A, é possível ver que existe uma reduzida área de sobreposição entre as duas imagens (devido ao deslocamento de uma das impressões muito para cima), em B, está um exemplo de deformação não-linear e finalmente em C, um exemplo de duas impressões que apresentam condições de captura bastante distintas.

---

<sup>1</sup>Leitores biométricos, são digitalizadores, desenhados para a captura de impressões digitais, com o objectivo de identificar um indivíduo. As técnicas utilizadas no processamento biométrico são, no seu todo, similares com as técnicas utilizadas neste estudo.

## 2.1. DACTILOSCOPIA

---



**Figura 2.6** Impressões digitais que apresentam dificuldades para o seu reconhecimento.



# Capítulo 3

## Estado da arte e aplicações

### 3.1 Estado da Arte

Não existe no universo das aplicações AFIS um grande número de soluções desenvolvidas, no entanto é possível encontrar algumas que funcionam de modo satisfatório. Neste capítulo é feita uma breve análise comparativa e descritiva de cada uma dessas aplicações.

O **eFinger**<sup>1</sup> é uma solução AFIS *open-source*. Os seus autores definiram como objectivos para a mesma fazer corresponder uma impressão digital introduzida pelo utilizador, com um conjunto de impressões digitais guardadas numa base de dados. A implementação da aplicação foi feita recorrendo à linguagem de programação C++ e ao programa comercial *Matlab*<sup>2</sup>, funcionando exclusivamente em ambientes *Windows*<sup>3</sup>.

A aplicação foi desenvolvida no âmbito de uma disciplina académica e é composta por diferentes módulos que fazem individualmente todo o processamento necessário para a identificação de uma impressão digital [10]. Esta implementa três algoritmos diferentes para efectuar a comparação das impressões digitais, sendo eles: *Relative Distance Matching*, *Quad Tree* e *Image Mapping*. Cada um destes algoritmos baseia-se numa pontuação resultante da comparação dos mapas das duas imagens (imagem a comparar com cada uma das imagens da base de dados). Consoante a pontuação, é possível aceitar a impressão digital como podendo ser igual, ou simplesmente rejeitá-la.

Em termos temporais, a aplicação demora aproximadamente 12 segundos a processar cada imagem antes de a adicionar à base de dados.

---

<sup>1</sup>Disponível para download em <http://sourceforge.net/projects/efinger/>

<sup>2</sup>Copyright The MathWorks

<sup>3</sup>Sistema operativo desenvolvido pela empresa Microsoft.



Algumas das limitações desta aplicação são a necessidade de utilizar imagens de boa qualidade e a falta de tratamento para as impressões digitais que apresentam rotações (a mesma imagem com um ângulo de captura diferente é identificada como sendo uma imagem diferente) [10].

O *Fingerprint Verification System (FVS)*<sup>4</sup> é também ele um sistema de verificação de impressões digitais *open-source* desenvolvido como trabalho de uma disciplina académica. Este foi criado com o objectivo de ser uma biblioteca que permitisse aos programadores adicionarem a tecnologia de processamento de impressões digitais aos seus programas.

Os objectivos principais definidos para a aplicação foram a simplicidade, reduzido tamanho, desempenhos de velocidade elevados e ser multi-plataforma [11].

Devido ao facto da aplicação não implementar os métodos necessários para trabalhar com imagens, as mesmas têm de ser sujeitas a uma prévia preparação antes de poderem ser utilizadas pelo programa. Este processamento transforma a imagem numa estrutura composta por vectores e matrizes, que é posteriormente utilizada pelo programa nos processos de pré-processamento e comparação [12].

O *NIST Biometric Image Software (NBIS)*<sup>5</sup> é uma colecção de aplicações para processamento de impressões digitais. O seu desenvolvimento e manutenção está a cargo de um grupo do *National Institute of Standards and Technology (NIST)*<sup>6</sup>.

Este pacote foi desenvolvido para ser utilizado pelo *Federal Bureau of Investigation (FBI)* e pelo *Department Of Homeland Security (DHS)* [13]. O NBIS é utilizado no controle do crime, por parte do FBI e no controle às fronteiras, por parte do DHS.

Como o NBIS foi desenvolvido por empregados do governo dos Estados Unidos da América no decorrer das suas obrigações, o software não está sujeito a leis de direito de autor e é de domínio público [14].

O NBIS é também ele uma aplicação modular em que cada módulo funciona independentemente, mas os melhores resultados são obtidos quando executados sequencialmente.

A simples combinação de dois dos módulos permite determinar se duas imagens de impressões digitais correspondem ao mesmo dedo, um requisito obrigatório para qualquer sistema de identificação baseado em impressões digitais. Além de implementar um módulo capaz de classificar as impressões digitais (ver capítulo 4), o NBIS implementa também um detector de orientação das minúcias. Esta informação vai ser bastante útil no processo de comparação, uma

---

<sup>4</sup>Disponível para download em : <http://fvs.sourceforge.net/index.html>

<sup>5</sup><http://fingerprint.nist.gov/NBIS/index.html>

<sup>6</sup><http://www.nist.gov/index.html>

### 3.1. ESTADO DA ARTE

---

vez que, deste modo, é possível resolver os problemas das rotações das impressões digitais.

O *Verifinger*<sup>7</sup> é um pacote de desenvolvimento de software (SDK) proprietário que implementa diversos métodos necessários para a verificação de impressões digitais. É desenvolvido pela empresa *NeuroTechnology*<sup>8</sup>, e foi criado a pensar em programadores de sistemas biométricos.

Esta solução tem suporte para impressões digitais que apresentam deformações, translações ou rotações, assim como um grande aumento de desempenho resultante de uma prévia classificação das mesmas [15].

Uma das limitações do SDK é que este apenas trabalha com imagens que tenham no mínimo 250 pontos por polegada (DPI). Em termos temporais, apresenta um tempo de processamento médio de 0.3 segundos por impressão digital e consegue atingir velocidades de cerca de 40.000 impressões digitais por segundo [16].

A *Fingerprint Verification Competition*<sup>9</sup> (FVC) é a maior competição mundial destinada a softwares de identificação de impressões digitais [17].

É organizada de 2 em 2 anos pela Universidade de Bolonha (Itália), Universidade Estatal de São José (EUA), Universidade Estatal de Michigan (EUA) e pela Universidade Autónoma de Madrid (Espanha) [18].

Os objectivos principais desta competição são fomentar os avanços no campo dos sistemas biométricos, mas também manter registo dos avanços feitos na área da identificação de impressões digitais, tanto para fins académicos como comerciais [17].

A última edição da competição contou com um total de 53 participantes (27 empresas, 13 universidades e 13 programadores independentes) que submeteram um total de 70 algoritmos. O algoritmo vencedor desta edição foi submetido pela empresa Koreana, *Suprema Inc* e demorava aproximadamente 0,04 segundos a fazer uma comparação de uma impressão digital.

Em termos globais, todas estas soluções partilham entre si o mesmo conjunto de técnicas de pré-processamento, havendo apenas algumas diferenças em relação aos métodos utilizados para fazer a comparação das impressões digitais.

Comparando as soluções apresentadas entre elas, o NBIS é a solução mais completa, uma vez que conjuga todos os módulos necessários com um módulo de classificação de impressões digitais, tendo ainda suporte para impressões que apresentam rotações. Juntamente com estas

---

<sup>7</sup>Copyright NeuroTechnology

<sup>8</sup><http://www.neurotechnology.com/>

<sup>9</sup><http://bias.csr.unibo.it/fvc2006/>

qualidades, é ainda uma solução *open-source* podendo ser melhorada por qualquer pessoa. Em termos de velocidade de processamento, tendo em conta os dados disponíveis, a solução proprietária *Verifinger* é a que apresenta tempos mais rápidos.

### 3.2 Aplicações AFIS

Os sistemas automatizados de identificação de impressões digitais (AFIS) são maioritariamente utilizados por forças de segurança, e estas utilizam bases de dados de impressões digitais de elevadas dimensões [19]. Este facto, juntamente com o grande custo computacional exigido pelos algoritmos de verificação de impressões digitais [4], faz com que as técnicas de classificação sejam uma mais valia em termos de desempenho para estes sistemas.

#### 3.2.1 eFinger

O *eFinger* é uma aplicação modular que contém métodos para aperfeiçoamento de imagem, extracção de cristas, binarização, e outros necessários para a correcta comparação das impressões digitais [10].

O objectivo dos módulos de processamento de imagem é preparar a mesma de modo a que seja possível fazer a extracção das características com o menor número de erros.

São utilizados filtros para a remoção de ruído, para transformar a imagem original numa imagem com apenas pretos e brancos (binarização) e finalmente a esqueletização que transforma todas as linhas da impressão digital em linhas com apenas 1 pixel de largura (ver capítulo 5).

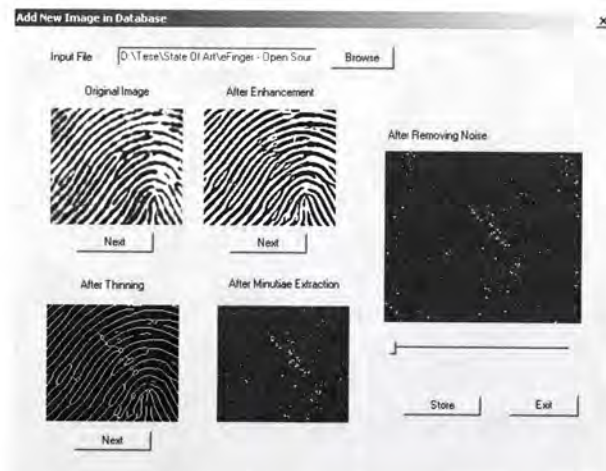


Figura 3.1 Resultado da aplicação dos filtros numa impressão digital nova.

## 3.2. APLICAÇÕES AFIS

Depois do aperfeiçoamento da imagem, já é possível fazer a extracção de minúcias que localiza todos os pontos característicos da impressão digital e o pós-processamento que remove todas as falsas minúcias introduzidas pelas técnicas de melhoramento da imagem.

Sempre que se pretende adicionar uma imagem à base de dados, esta é processada por todos estes módulos e só depois é guardada.

Na figura 3.1, podemos ver o resultado de introduzir uma imagem na base de dados, após a aplicação dos filtros que constituem a aplicação (binarização, esqueletização e extracção de minúcias).

Estes últimos módulos geram um mapa do posicionamento das minúcias e é este que é gravado na base de dados e servirá posteriormente para a comparação.

Na figura 3.2, é possível ver o resultado de uma comparação utilizando o algoritmo *Relative Distance Matching*.

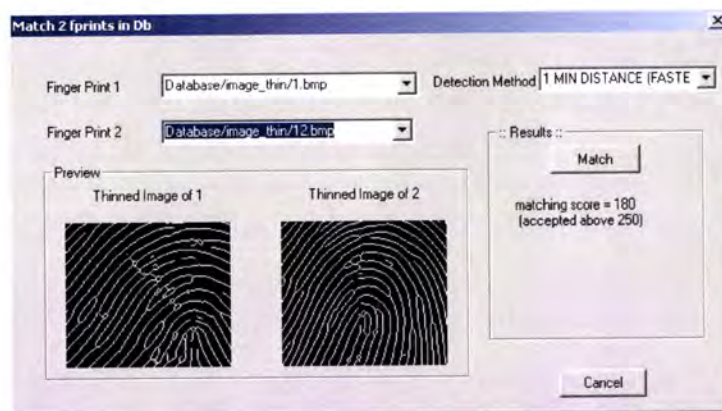


Figura 3.2 Resultado da comparação de duas impressões digitais.

### 3.2.2 Fingerprint Verification System

O FVS partilha também com as restantes aplicações um conjunto de técnicas de processamento de imagem com vista a melhorar a mesma para o processo de extracção de características.

A extracção de minúcias é feita através da comparação de cada pixel e seus vizinhos com matrizes que representam os padrões correspondentes às características a extrair.

A forma de comparação das impressões digitais é dividida em dois processos. Primeiro o programa tenta sobrepor as duas imagens, movimentando-as verticalmente e horizontalmente de modo a que a maioria dos pixels se sobreponham. Se uma determinada percentagem das linhas ficarem sobrepostas, então procede-se para o passo seguinte. No passo seguinte é então utilizada a informação das características extraídas. Para cada uma das minúcias das duas

imagens é comparada a sua localização, tipo de minúcia e o ângulo feito pela crista. A impressão digital é aceite como sendo igual apenas se todas estas comparações tiverem um resultado positivo [20].

### 3.2.3 NIST Biometric Image Software

Como dito em 3.1, o NBIS é uma aplicação composta por diversos módulos que quando utilizados sequencialmente apresentam os melhores resultados. Os módulos que constituem a aplicação são os seguintes:

- PCASYS

O PCASYS é o módulo de classificação das impressões digitais. Este módulo permite sub-dividir as impressões em categorias, podendo reduzir bastante o tempo de pesquisa (ver capítulo 4).

- NFIQ

O NFIQ é o módulo composto pelos algoritmos de aperfeiçoamento de imagens. Este módulo é bastante importante uma vez que a qualidade da imagem tem papel determinante nos resultados da comparação. Sempre que se utilizam imagens com baixa qualidade, o desempenho dos algoritmos tende a diminuir [13].

- MINDTCT

O MINDTCT é o módulo que faz a detecção e extracção das minúcias. Para cada minúcia extraída guarda na base de dados a sua localização, orientação, tipo e qualidade.

- BOZORTH3

Finalmente, o pacote inclui ainda o módulo BOZORTH3 que é o responsável por fazer a comparação das impressões.

O BOZORTH3 foi desenvolvido originalmente por Allan S. Bozorth no FBI, com o objectivo de investigar algoritmos com uma possível solução para os problemas das translações e rotações das impressões digitais. Os investigadores do NIST melhoraram-no e integraram-no no seu pacote [13].

O processo de comparação é feito através da análise de dois mapas de minúcias de duas imagens diferentes. O programa avalia a sua similaridade e devolve uma pontuação como saída. Após isto, é definido um limiar aceitável, que é comparado com a pontuação final, de modo a decidir, se os dois mapas correspondem ao mesmo dedo [13].

## 3.2. APLICAÇÕES AFIS

### 3.2.4 Verifinger

O SDK *Verifinger* é composto por vários algoritmos proprietários que utilizam, assim como todas as soluções aqui apresentadas, os pontos característicos (minúcias) das impressões digitais de modo a efectuar as comparações. A figura 3.3, ilustra uma demonstração do SDK.



**Figura 3.3** Inserção de uma imagem na base de dados utilizando o Verifinger.

### 3.2.5 Fingerprint Verification Competition

Como dito anteriormente, a FVC é uma competição de algoritmos para comparação de impressões digitais. Para testar os algoritmos concorrentes, são utilizadas 4 bases de dados de impressões digitais obtidas de diferentes maneiras. Na figura 3.4, é possível observar alguns exemplos de impressões digitais utilizadas na competição.

A competição funciona em duas modalidades : *Open* e *Light*. Na modalidade *Light*, os algoritmos têm fortes restrições no tempo de execução e no espaço necessário para armazenar as características extraídas das imagens. Na modalidade *Open*, as restrições são menores, mas sempre garantindo que o algoritmo possa ser utilizado de forma comercial [17].

A competição baseia-se em dois tipos de avaliações para os algoritmos participantes: resultados médios, que são calculados a partir de todas as imagens em todas as bases de dados; resultados por base, que são calculados em cada base de dados de maneira individual.

Os resultados médios medem a robustez do algoritmo quando aplicado a imagens de sensores com qualidade e características diferentes. Tentam descrever como o algoritmo resolve o problema da verificação de impressões digitais de maneira universal, ou seja, como ele se comporta com impressões digitais de diferentes origens. Os resultados por base descrevem como o algoritmo se comporta para cada tipo de impressão digital de maneira independente, o que permite aos participantes afinarem o mesmo.

São utilizadas diferentes medidas para a avaliação do desempenho dos algoritmos, sendo a mais importante, a taxa de erro de comparação (EER). Esta medida mede a probabilidade do



**Figura 3.4** Alguns exemplos de impressões digitais utilizadas na competição.

algoritmo errar ao decidir se duas imagens correspondem entre si, isto é, mede a taxa de erro de comparação [21].

Como as imagens utilizadas são obtidas de diferentes fontes, as mesmas podem apresentar características muito diferentes. Este facto pode originar que um algoritmo atinja um EER muito baixo para determinadas imagens e muito alto para outras. Assim é utilizado o EER médio, que pode ser interpretado como a taxa de erro de comparação média, isto é, a probabilidade média do algoritmo errar na comparação de duas impressões digitais de diferentes origens.

# Capítulo 4

## Metodologias de Classificação

O primeiro sistema de regras de classificação foi proposto em 1823 por Purkinje que dividiu as impressões digitais em nove categorias de acordo com a configuração total das cristas papilares das mesmas. O primeiro estudo aprofundado sobre o tema foi feito por Francis Galton que reduziu esse grupo para três classes principais: arcos, laços e verticilos. Mais tarde o próprio dividiu ainda cada uma das classes em subcategorias. Em 1902, Edward Henry modificou o sistema de classificação de Galton aumentando-lhe o número de classes para 5 [4].

### 4.1 Sistema de Classificação de Henry

O sistema de classificação de Henry divide as impressões digitais nas seguintes classes:

- Laços

Uma impressão digital desta classe apresenta pelo menos uma crista que começa de um dos lados, faz uma curva e termina do mesmo lado onde começou (daqui vem a designação de laço). Existe pelo menos um laço e um delta. O delta assume-se que fica a sul do laço. Os laços podem ser subdivididos em duas categorias: laços esquerdos e laços direitos. A designação dá-se, respectivamente, às cristas que começam e acabam do lado esquerdo e as que começam e acabam do lado direito.

- Arco plano

Não apresenta deltas nem laços. As cristas papilares atravessam o dedo de um lado ao outro podendo formar pequenas elevações.

- Arco angular

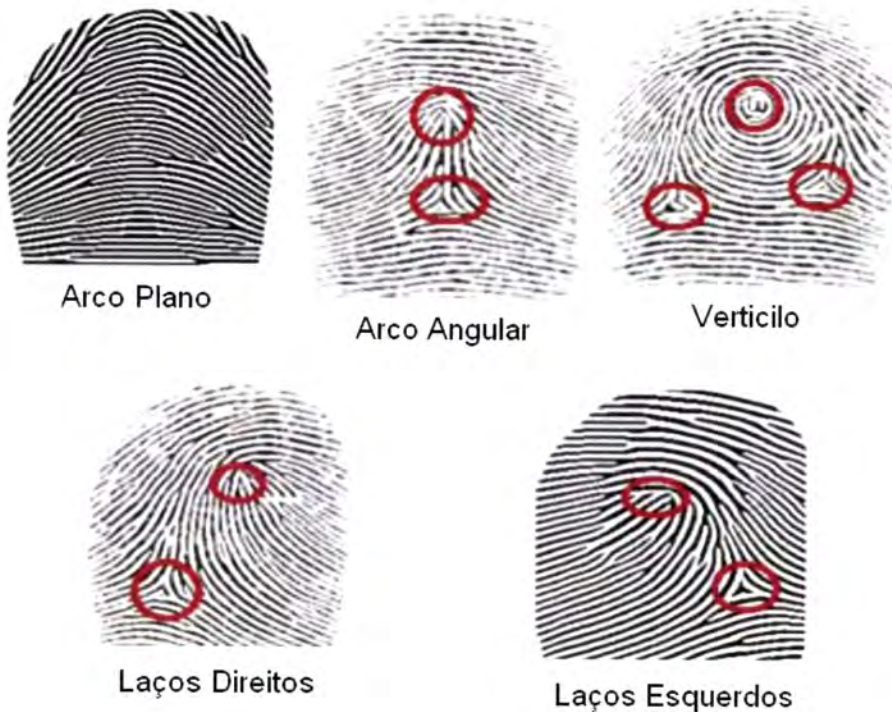
Semelhante ao arco plano, mas pelo menos uma das cristas apresenta uma grande curvatura. Existe também um laço e um delta.



- Verticilos

Uma impressão digital desta classe apresenta pelo menos uma crista que completa um caminho de  $360^\circ$  à volta do centro da impressão digital. Nestas condições encontram-se impressões digitais que podem apresentar dois laços e dois deltas.

A figura 4.1 ilustra as classes do sistema de classificação de Henry.



**Figura 4.1** Classes do sistema de classificação de Henry. [Figura de [22]]

Ao serem aplicadas técnicas de classificação sobre uma base de dados de impressões digitais, permite-se que os algoritmos de verificação sejam aplicados apenas a um sub-conjunto da mesma. Segundo os arquivos do *Federal Bureau of Investigation* (FBI), 60% dos padrões são arcos, 30% são verticilos e 5% são laços. Por este motivo, as bases de dados de impressões digitais, que foram previamente classificadas, não possuem impressões digitais distribuídas uniformemente pelas 5 classes [22].

## 4.2 Métodos de classificação

Durante décadas, foram desenvolvidos um número considerável de métodos [22], com o objectivo de classificar as impressões digitais, dos quais se faz uma breve descrição a seguir:

## 4.2. MÉTODOS DE CLASSIFICAÇÃO

---

- **Análise estrutural**

Neste método é feita uma análise da configuração global dos padrões das impressões digitais. Nessa análise, está descrita a distribuição das direcções das cristas papilares e determina-se a existência dos pontos singulares. De acordo com o número de núcleos e deltas encontrados e respectiva localização, a imagem da impressão digital pode ser classificada numa das classes sugeridas pelo sistema de classificação de Henry [8].

- **Método estatístico**

Neste método, tal como o próprio nome sugere, são utilizadas técnicas estatísticas. É comum a utilização de vectores de características e de atributos espaciais (distâncias e ângulos das linhas) [4] para desenvolver uma visão geométrica de um padrão. As características estatísticas são calculadas com base nas cristas papilares e nos pontos singulares e são posteriormente utilizadas para decidir a classe a que a impressão digital corresponde [23].

- **Método sintáctico**

Neste método, é utilizado um conjunto de regras para representar e classificar os padrões das impressões digitais. É necessária a aplicação de uma linguagem formal para a descrição e reconhecimento dos padrões associados às imagens. O método sintáctico é baseado na descoberta das continuidades e paralelismos das cristas papilares, tipos de linhas, pequenos graus de variação local, núcleos e deltas. A representação é feita sobre a forma de vectores que podem ser comparados de forma a determinar a classe da impressão digital. Um dos principais problemas de classificação por meio de métodos sintácticos é a variedade de conjuntos de padrões, fazendo com que cada investigador construa o seu conjunto de regras para a sua aplicação [24, 25].

- **Redes Neurais Artificiais**

Um sistema automático de aprendizagem utiliza as direcções das cristas e outros aspectos da imagem da impressão digital como atributos de treino, de modo a diferenciar as várias classes. Nos últimos anos, as redes neurais artificiais têm vindo a destacar-se nos processos de classificação, identificação de minúcias e reconhecimento de impressões digitais [26].

- **Métodos híbridos**

Os métodos híbridos são aqueles em que se conjugam pelo menos dois dos métodos acima citados para se fazer a classificação de uma impressão digital. Nos artigos consultados, verificou-se que em [27, 28], os autores combinam os métodos estatísticos e estrutural.

### 4.3 Técnicas de Classificação

O processo de classificação de impressões digitais, pode ser dividido em 3 etapas distintas:

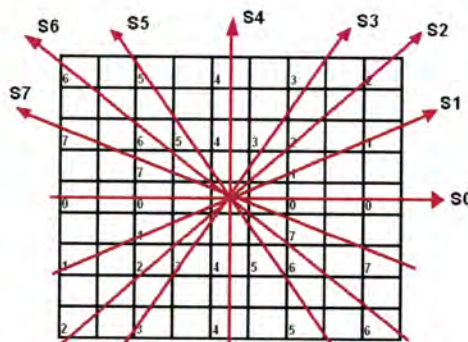
- Cálculo do mapa direccional.
- Suavização do mapa direccional.
- Cálculo do índice de Poincaré.

Este processo, tem como objectivo assegurar que uma dada impressão digital, pertence a uma das diferentes classes propostas por Henry (ver capítulo 4.1), de acordo com as suas propriedades geométricas. A classificação de uma impressão digital, antes de ser guardada numa base de dados, permite posteriormente reduzir o tempo de pesquisa.

#### 4.3.1 Cálculo do mapa direccional

O mapa direccional de uma impressão digital, corresponde a uma representação vectorial, da direcção estimada, de um conjunto de pixels. Através deste mapa, é possível obter informações, que se encontram contidas nos padrões das impressões digitais. Este método, pode também ser utilizado, com bons resultados, em imagens que apresentem grande quantidade de ruído, uma vez que a influência do mesmo, pode ser reduzida através do cálculo da média das direcções. Por este motivo, a maioria dos métodos de classificação, são baseados neste mapa direccional [29, 19].

Em [29, 26], é proposto um método para o cálculo do mapa direccional de um conjunto de pixels, que recorre a uma máscara 9x9 (figura 4.2), centrada no pixel de interesse.



**Figura 4.2** Mascara 9x9 para o cálculo do mapa direccional.

### 4.3. TÉCNICAS DE CLASSIFICAÇÃO

---

Numa imagem de uma impressão digital, as cristas são representadas por tons mais escuros, e os vales por tons mais claros. Deste modo, utiliza-se os valores das intensidades dos níveis de cinzentos, no cálculo das direcções.

O valor dos tons de cinza da imagem na posição  $(i, j)$ , é representada por  $I(i, j)$ , e é utilizado no cálculo das grandezas  $S_0, S_1, \dots, S_7$ , que são obtidas através da aplicação das equações (4.1) a (4.8).

$$S(0) = \sum_{K=-2}^2 I(i, j+2K) - I(i, j) \quad (4.1)$$

$$S(1) = \sum_{K=-2}^2 I(i+K, j-2K) - I(i, j) \quad (4.2)$$

$$S(2) = \sum_{K=-2}^2 I(i+2K, j-2K) - I(i, j) \quad (4.3)$$

$$S(3) = \sum_{K=-2}^2 I(i+2K, j-K) - I(i, j) \quad (4.4)$$

$$S(4) = \sum_{K=-2}^2 I(i+2K, j) - I(i, j) \quad (4.5)$$

$$S(5) = \sum_{K=-2}^2 I(i+2K, j+K) - I(i, j) \quad (4.6)$$

$$S(6) = \sum_{K=-2}^2 I(i+2K, j+2K) - I(i, j) \quad (4.7)$$

$$S(7) = \sum_{K=-2}^2 I(i+K, j+2K) - I(i, j) \quad (4.8)$$

Após o cálculo das grandezas  $S_0, S_1, \dots, S_7$ , é então possível, utilizar esses valores, para definir a direcção do pixel.



Sejam  $p$  e  $q$ , dentro do intervalo  $[0,7]$ , os índices das equações (4.9) e (4.10). Os valores de  $p$  e  $q$ , vão ser, respectivamente, os valores dos índices, do mínimo e máximo de  $S_i$ , ou seja, se  $S_3$  for o mínimo, e  $S_7$  o máximo, então,  $p = 3$  e  $q = 7$ .

A direcção  $d$ , será  $p$ , se o pixel central estiver localizado numa crista (valores de intensidade de cinzento mais altos), e  $q$  se estiver localizado num vale (valores de intensidade de cinzento mais baixos).

$$Sp = \min(S_i) , \text{ com } i=0\dots7 \quad (4.9)$$

$$Sq = \max(S_i) , \text{ com } i=0\dots7 \quad (4.10)$$

Designando como  $V$  o valor da intensidade do pixel central, a sua direcção final, é dada pela equação 4.11.

$$d = \begin{cases} p & \text{se } (4V + Sp + Sq) < \frac{3}{8} \sum S_i, \text{ com } i=0\dots7 \\ q & \text{caso contrário.} \end{cases} \quad (4.11)$$

A equação 4.11, calcula a direcção de cada pixel, nas 8 direcções possíveis. O mesmo método, pode ser utilizado para o cálculo das 16 direcções, sendo apenas necessário utilizar uma máscara  $17 \times 17$ , e calcular as restantes grandezas  $S_8, \dots, S_{15}$  [22, 30].

### 4.3.2 Suavização do mapa direccional

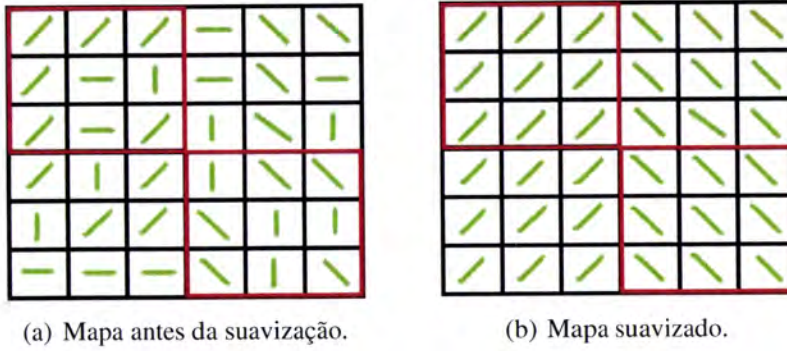
Após o cálculo do mapa direccional, é necessário proceder a uma suavização da mesma, devido a possíveis erros causados por ruído. Em [22, 31], é apresentado um método de suavização do mapa direccional, utilizando o método estatístico moda.

Para utilizar este método, é necessário dividir o mapa em blocos. Para cada bloco seleccionado, é calculada a direcção que ocorre com maior frequência, sendo esta, depois aplicada a todo o bloco. A figura 4.3 demonstra a utilização deste processo, num mapa direccional de pequenas dimensões. Como se pode observar, na matriz resultante da suavização, a cada bloco  $3 \times 3$  foi aplicada a direcção com maior frequência.

A figura 4.4<sup>1</sup> apresenta o resultado final da aplicação deste método, a uma imagem de uma

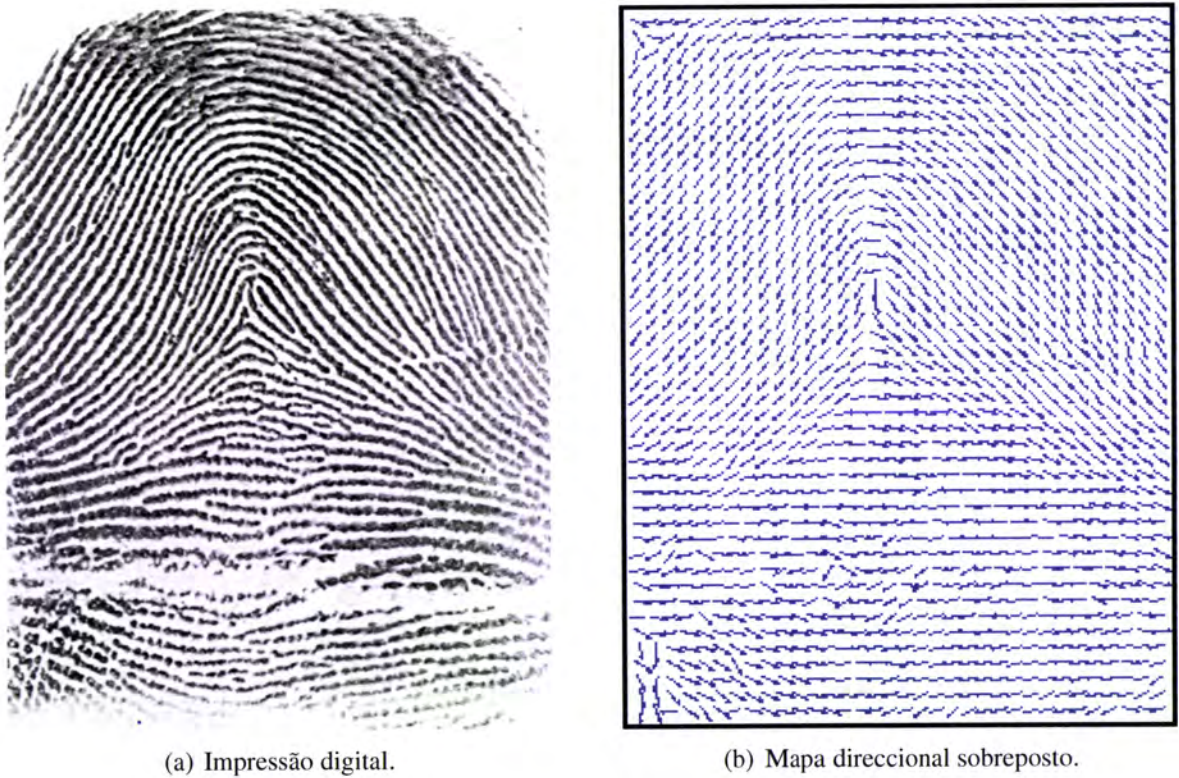
<sup>1</sup>A imagem do mapa direccional apresentada, foi obtida através da aplicação da função *plotridgeorient*, desenvolvida por Peter Kovési, e disponível para download em <http://www.csse.uwa.edu.au/pk/Research/MatlabFns/FingerPrints/plotridgeorient.m>.

### 4.3. TÉCNICAS DE CLASSIFICAÇÃO



**Figura 4.3** Exemplo do método de suavização.

impressão digital, de grandes dimensões.



**Figura 4.4** Mapa direccional da impressão digital.

Como se pode observar, o mapa direccional, com excepção de algumas situações, acompanha rigorosamente as linhas da impressão digital.

### 4.3.3 Classificação pelo índice de Poincaré

Tal como descrito em 2.1.1, os núcleos e os deltas identificados numa impressão digital, são utilizados para classificar a mesma. O índice de Poincaré, tem como objectivo a localização destes pontos na imagem.

Para cada pixel do mapa direccional, o índice de Poincaré classifica o mesmo como sendo ordinário (sem interesse), núcleo ou delta [22].

O índice de Poincaré, é calculado sobre uma curva fechada em redor de um pixel. O índice, é calculado, através do somatório das diferenças, dos valores dos ângulos do mapa direccional, adjacentes a essa curva, no sentido anti-horário. O resultado deste cálculo, é sempre um valor de  $0^\circ$  (pixel ordinário),  $180^\circ$  positivos (núcleo) ou  $180^\circ$  negativos (delta). A fórmula matemática do cálculo do índice de Poincaré, é dada pela equação (4.12), onde  $(i, j)$  representa as coordenadas do ponto a analisar, e  $A_k$ , o valor do ângulo de cada ponto que o circula.

$$P(i, j) = \sum_{k=1 \dots 4} (A_{(k+1) \bmod 4} - A_k) \quad (4.12)$$

A diferença entre cada 2 ângulos é calculada considerando uma vizinhança de 2x2 pixels [30, 32]. No caso do valor calculado ser superior a  $90^\circ$ , é necessário verificar o sinal. No caso do sinal ser negativo, soma-se  $180^\circ$  ao valor, caso contrário, subtrai-se  $180^\circ$  [22].

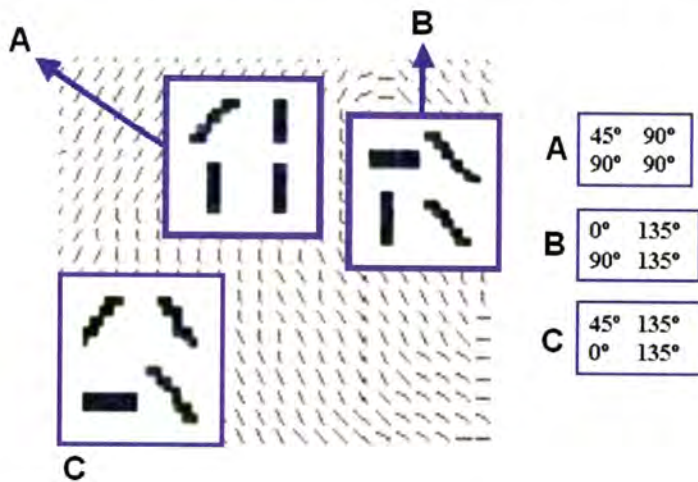


Figura 4.5 Exemplo de cálculo do índice de Poincaré. [Figura de [22]]

Utilizando os valores sugeridos na imagem 4.5, o índice de Poincaré, é calculado respectivamente para A, B e C, da seguinte maneira:

No cálculo de B, foi necessário adicionar  $180^\circ$ , uma vez que uma das diferenças era superior a  $90^\circ$ , e o sinal era negativo. No cálculo de C, verifica-se exactamente o contrário. Uma das

### 4.3. TÉCNICAS DE CLASSIFICAÇÃO

$$\begin{aligned} A & (90-45)+(90-90)+(90-90)+(45-90)=45-45=0 \\ B & (90-0)+(135-90)+(135-135)+(0-135)=90+45+(-135+180)=+180 \\ C & (0-45)+(135-0)+(135-135)+(45-135)= -45+(135-180)+(-90)=-180 \end{aligned}$$

Figura 4.6 Cálculo matemático do índice de Poincaré.

diferenças era superior a  $90^\circ$ , mas o seu sinal era positivo. Deste modo, é necessário somar  $180^\circ$ . Com estes valores, é então possível afirmar que:

- A é um ponto ordinário.
- B é um núcleo.
- C é um delta.

Com os resultados obtidos através do índice de Poincaré, é então possível classificar uma impressão digital, numa das classes de Henry. O fluxograma 4.7 ilustra o processo de classificação, em cada uma das classes.

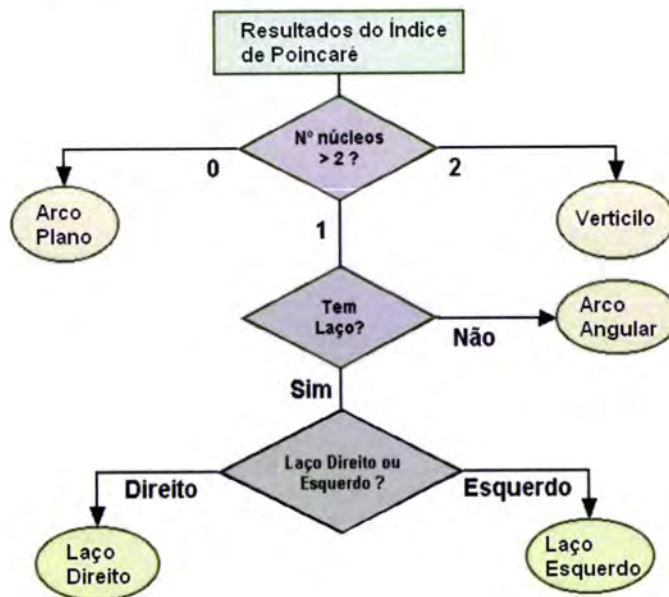


Figura 4.7 Classificação utilizando o índice de Poincaré. [Figura adaptada de [22]]

Para diferenciar os laços esquerdos dos laços direitos, é necessário observar a posição relativa do delta, a partir do núcleo. Se o ponto delta estiver do lado esquerdo do núcleo, a



impressão digital pertence à classe laço esquerdo, caso contrário, pertence à classe laço direito. De outro modo, se o ângulo formado entre os pontos núcleo e delta, for inferior a  $80^\circ$ , o delta está à esquerda, logo pertence à classe laço esquerdo. No caso de o ângulo formado entre os dois pontos for superior a  $100^\circ$ , o delta está do lado direito, logo pertence à classe laço direito [22].

### 4.4 Origem do código utilizado

O código fonte utilizado, na implementação do sistema de classificação, na aplicação desenvolvida, faz parte da aplicação Gaudi-Afis.

O Gaudi-Afis, é um sistema de classificação, dividido em módulos independentes. Cada um desses módulos é responsável por cada uma das partes da classificação. Foi desenvolvida na linguagem de programação C, por Murilo Lima.

O módulo que faz o cálculo da imagem direccional utilizado, foi desenvolvido em *Octave*, por Peter Kovesi.

Para fazer a leitura e escrita de imagens, é utilizada como biblioteca, a aplicação de código aberto *ImageMagick*<sup>2</sup>.

---

<sup>2</sup><http://www.imagemagick.org/script/index.php>

## Capítulo 5

# Técnicas para o Processamento de Impressões Digitais

A construção de um sistema de identificação através de impressões digitais passa pela implementação de um conjunto de técnicas, que tentam tornar o sistema o menos falível possível. No entanto, não existe ainda, um conjunto pré-definido de técnicas, que tornem estes sistemas infalíveis.

Na literatura [33, 34, 35, 36, 37], é possível encontrar estudos, que se baseiam apenas em técnicas de binarização e esqueletização, aplicando de seguida, métodos para a extracção de minúcias. No entanto, uma vez que as imagens de impressões digitais, normalmente são de má qualidade, a utilização de apenas estas abordagens, não é suficiente. Em [30, 38, 39, 40, 41, 42, 43, 44], são propostas outras técnicas (tais como filtros de remoção de ruído e algoritmos para o melhoramento das cristas), que têm como objectivo o melhoramento das imagens de impressões digitais, com vista a um melhor resultado, na altura da extracção das características.



**Figura 5.1** Diagrama do sistema.

A implementação de um sistema AFIS, pode ser dividida em pré-processamento, extracção de características e pós-processamento, sendo cada uma delas constituída por diferentes técnicas.

Na figura 5.1, é possível ver a sequência de acções propostas, desde a selecção da impressão digital, até à sua introdução na base de dados.

O processo de classificação descrito em 4, e presente na figura 5.1, possibilita a separação das representações das impressões digitais, em classes, na base de dados. Esta separação permite uma diminuição do espaço de pesquisa, no momento da comparação de impressões digitais.

### 5.1 Pré-processamento

Devido à má qualidade que as imagens de impressões digitais, frequentemente apresentam, é necessário aplicar um conjunto de técnicas, que tentem melhorar um pouco, os aspectos importantes a extrair das mesmas. Assim, a implementação de um sistema AFIS, começa pelo pré-processamento das imagens.

A fase do pré-processamento, tem como objectivo a melhoria das imagens das impressões digitais, como tal, inclui normalização de contraste, filtro de mediana, algoritmos para melhoramento das cristas, binarização e esqueletização [43].

Ao aplicar estes métodos sobre a imagem original, obtêm-se uma nova imagem, sobre a qual é possível aplicar os algoritmos de extracção de características, de uma maneira mais simples e mais fiável.

#### 5.1.1 Normalização do contraste

As imagens de impressões digitais são na maioria das vezes, de má qualidade, e têm deficiências nos níveis de contraste. Por este motivo, podem originar combinações de pixels, que originem a extracção de falsas minúcias.

A normalização do contraste de uma imagem, tem como objectivo, obter o máximo da variância do histograma da mesma. Deste modo, consegue-se o melhor nível de contraste para a imagem de entrada. Esta normalização aproxima o histograma da imagem de entrada para um histograma uniforme [45, 46].

Antes de ser aplicada a normalização, é necessário definir os valores mínimos e máximos para cada pixel. Uma vez que as imagens em tons de cinzentos apresentam valores entre 0 e 255, são estes os valores utilizados, os quais são designados respectivamente por  $p_{min}$  e  $p_{max}$ .

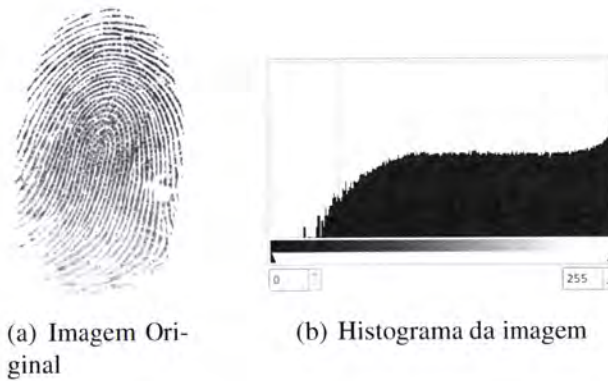
Uma vez definidos os limites, é necessário percorrer toda a imagem, guardando o valor mínimo e máximo de intensidade, presentes na mesma. Estes valores são designados por  $imin$  e  $imax$ .

## 5.1. PRÉ-PROCESSAMENTO

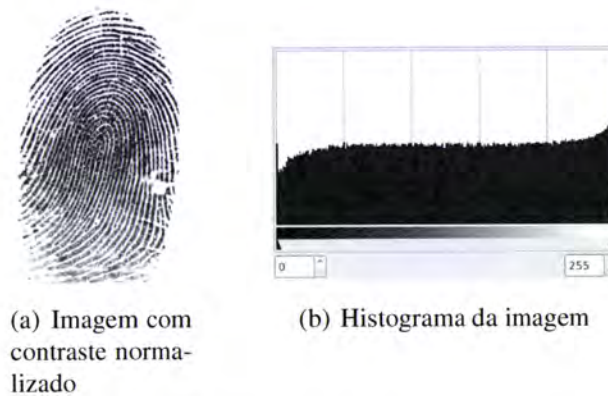
Após a definição dos valores anteriores, a normalização do contraste de uma imagem, é feita aplicando a fórmula 5.1, a cada pixel da mesma, sendo  $P_{in}$  o pixel a normalizar e  $P_{out}$  o pixel normalizado [47].

$$P_{out} = (P_{in} - i_{min}) \left( \frac{p_{max} - p_{min}}{i_{max} - i_{min}} \right) + p_{min} \quad (5.1)$$

Nas imagens 5.2 e 5.3, é possível ver uma impressão digital e seu respectivo histograma, antes e depois da normalização do contraste. Como se pode observar, pela análise do histograma, após a normalização do contraste, os valores dos níveis de cinza que compõem a imagem, encontram-se agora, aproximadamente uniformizados, pelos 255 valores possíveis.



**Figura 5.2** Impressão digital e seu histograma antes da normalização do contraste.



**Figura 5.3** Impressão digital e seu histograma após a normalização do contraste.

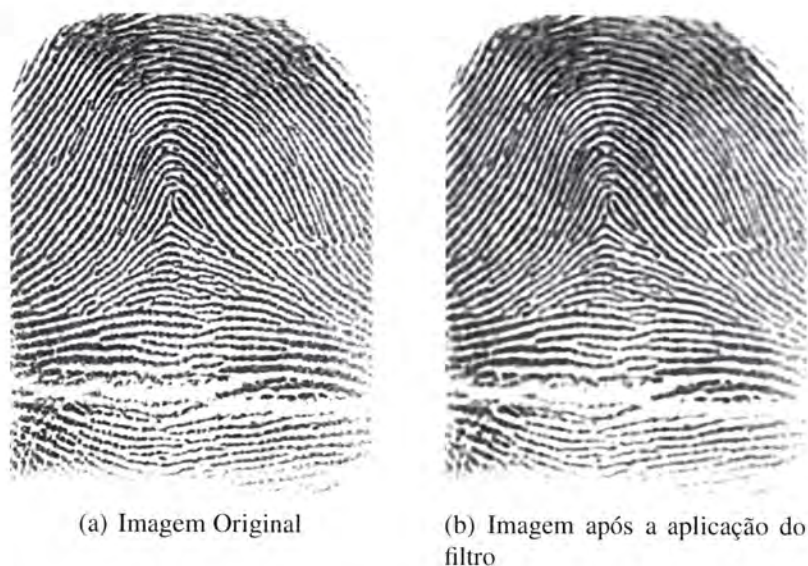
### **5.1.2 Filtro de Mediana**

Uma vez que as imagens de impressões digitais apresentam frequentemente bastante ruído, e a sua presença pode originar a extracção de falsas minúcias, é necessário aplicar filtros que o tentem diminuir.

O filtro de mediana aplicado, tem como objectivo, atenuar o ruído existente nos vales das impressões digitais, de modo a melhorar a diferenciação entre estes e as cristas das impressões digitais.

O filtro consiste em modificar o valor de um pixel, pelo valor da mediana da sua vizinhança. Para tal, é utilizada uma vizinhança  $N \times N$ , na qual os valores dos seus pixels, são ordenados de forma crescente de intensidade. Após a ordenação das intensidades, é seleccionado o valor central (mediana), e esse valor é então aplicado ao pixel de interesse.

Este filtro é bastante eficiente, para ruídos com intensidades bastante diferentes, das intensidades da sua vizinhança [48].



**Figura 5.4** Impressão digital antes e depois do filtro de mediana.

A imagem 5.4, mostra duas impressões digitais. À esquerda, a imagem original com ruído, e à direita, a mesma impressão digital após a passagem pelo filtro de mediana. Através da observação das duas imagens, é possível encontrar uma elevada diminuição do ruído existente.

### **5.1.3 Melhoramento das cristas**

Os algoritmos de melhoramento das cristas, têm como objectivo realçar as linhas que constituem as impressões digitais, uma vez que as mesmas, aquando da sua captura, sob a forma

## 5.1. PRÉ-PROCESSAMENTO

---

de imagem, podem ficar pouco nítidas e até mesmo apresentar manchas, que dificultam a sua correcta observação. Estes problemas podem levar ao aparecimento de erros, no momento da extracção dos pontos característicos.

Assim, são apresentados dois métodos, que têm como objectivo a recuperação e reconstrução das cristas das impressões digitais.

### 5.1.3.1 Root Filtering

Esta técnica de melhoramento das cristas das impressões digitais, permite que as cristas, que foram separadas devido à existência de ruído, sejam novamente ligadas.

A aplicação do algoritmo, é feita em blocos de  $M \times N$  pixels, aos quais é, em primeiro lugar, aplicada a transformada de *Fourier* (FFT)(5.2), onde  $M$  e  $N$  correspondem ao tamanho do bloco considerado (no caso do filtro em questão,  $M=32$  e  $N=32$ ) [49].

A *FFT* é uma ferramenta bastante utilizada no processamento de imagens, uma vez que permite a mudança do domínio espacial, para o domínio da frequência (onde o processamento é bastante mais rápido e mais simples, para certas operações), e vice-versa, sem haver perda de informação [30].

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} I(x, y) e^{-i2\pi * (\frac{ux}{M} + \frac{vy}{N})} \text{ com } u=0, \dots, M-1 \text{ e } v=0, \dots, N-1. \quad (5.2)$$

Este filtro, tem como objectivo reforçar as frequências dominantes de cada bloco, atenuando as mais fracas. Deste modo, para fazer o reforço das frequências mais dominantes, basta multiplicar a FFT de cada bloco, pelas suas magnitudes, elevadas a um valor  $k$  (5.3) [50, 39].

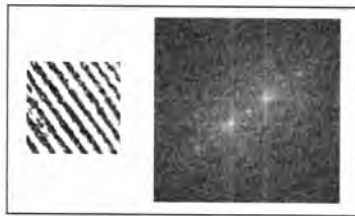
$$g(u, v) = |F(u, v)|^k \exp(i\angle F(u, v)) \quad (5.3)$$

O valor  $k$ , na equação (5.3), é uma constante determinada de forma experimental. Em [39], após testes exaustivos, chegou-se à conclusão que o valor óptimo de  $k$  é 0.45. Se o valor de  $k$  for maior, o filtro provoca deformações e quebras nas linhas [39].

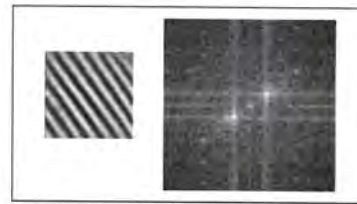
Finalmente, para se poder continuar a trabalhar com a imagem, é necessário aplicar a FFT inversa(5.4) a cada bloco. Deste modo, faz-se a passagem do domínio da frequência, para o domínio espacial novamente.

$$I(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} F(u, v) e^{i2\pi * (\frac{ux}{M} + \frac{vy}{N})} \text{ com } x=0, \dots, M-1 \text{ e } y=0, \dots, N-1. \quad (5.4)$$

A figura 5.5, mostra dois blocos de 32 pixels, e respectivos espectros de *Fourier*, antes e após a aplicação do filtro.



(a) Bloco e espectro de *Fourier* antes da aplicação do filtro.



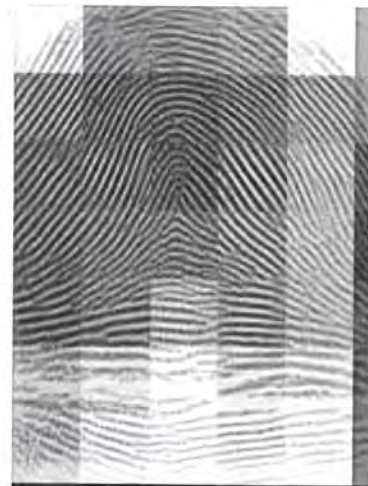
(b) Bloco e espectro de *Fourier* depois da aplicação do filtro.

**Figura 5.5** Blocos de uma impressão digital antes e depois da aplicação do filtro.

A figura 5.6 mostra o resultado da aplicação do filtro, a uma impressão digital, após o seu processamento bloco a bloco.



(a) Imagem Original.



(b) Imagem filtrada.

**Figura 5.6** Impressão digital após a aplicação do *Root Filtering*.

Observando as imagens, verifica-se que, não só as linhas ficam mais nítidas, como também houve uma considerável redução do ruído existente.

## 5.1. PRÉ-PROCESSAMENTO

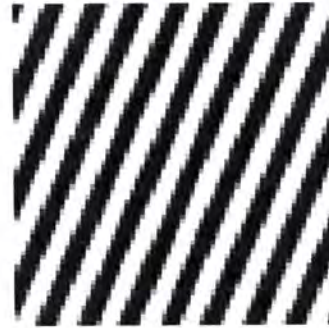
### 5.1.3.2 Filtros de Gabor

Um filtro de *Gabor*, é um filtro linear, definido como uma função harmónica, multiplicada por uma gaussiana. A função harmónica pode ser seno, co-seno, ou uma junção das duas, na forma de uma exponencial complexa [51, 43].

Os filtros de *Gabor*, são usualmente utilizados, em imagens que apresentam ondas sinusóides, como é o caso das impressões digitais [51, 52, 30]. É possível observar este comportamento, nas impressões digitais, em pequenos fragmentos da mesma (figura 5.7).



(a) Fragmento de uma impressão digital.



(b) Sinusóide com a mesma direcção da impressão digital.

**Figura 5.7** Fragmento de impressão digital e sinusóide com a mesma direcção.

A forma geral do filtro de *Gabor* é dada pela equação (5.5), onde  $x'$  e  $y'$  são dados, respectivamente, por (5.6) e (5.7).

$$G(x, y : \theta, f) = e^{(-\frac{1}{2}) * \left( \frac{x'^2}{\sigma_x^2} + \frac{y'^2}{\sigma_y^2} \right)} * \cos(2\pi f x') \quad (5.5)$$

Na referida equação (5.5),  $x$  e  $y$  representam as coordenadas espaciais na imagem,  $f$  a frequência da onda (corresponde ao mapa de frequências, calculado a seguir),  $\theta$  a orientação do filtro (para a orientação do filtro, é utilizado o mapa direccional, calculado em 4.3.1), e  $\sigma_x$  e  $\sigma_y$  o desvio padrão da função gaussiana, ao longo dos eixos  $x$  e  $y$  respectivamente [30].

$$x' = x \cos(\theta) + y \sin(\theta) \quad (5.6)$$

$$y' = -x \sin(\theta) + y \cos(\theta) \quad (5.7)$$



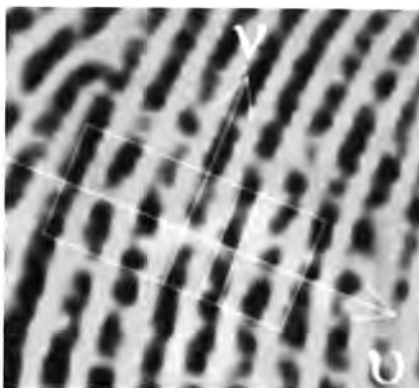
A correcta utilização do filtro de *Gabor*, na totalidade da impressão digital, depende de parâmetros, uma vez que as linhas das impressões digitais não têm todas a mesma direcção, e podem apresentar diferentes frequências ao longo da imagem. Deste modo, é necessário calcular, para cada imagem de impressão digital, o mapa de direcções (do mesmo modo que em 4.3.1) e a frequência espacial (corresponde à distância entre as cristas) [53].



**Figura 5.8** Diferenças de espaçamento entre as cristas, ao longo da imagem.

O cálculo da frequência envolve um conjunto de passos, que são explicados de seguida. Tomando como  $I$  a imagem original, divide-se a mesma em blocos de tamanho  $W \times W$  (no filtro em questão utilizam-se blocos  $16 \times 16$ ).

Para cada bloco, é criada uma janela de dimensões  $L \times W$  (ver figura 5.9), centrada num pixel  $(i, j)$  e orientada pela direcção do mesmo (mais uma vez, para o filtro em questão, utiliza-se uma janela de dimensões  $32 \times 16$ ), e calcula-se um vector designado por *x-signature*, de acordo com a equação 5.8 [53, 43].



**Figura 5.9** Janela orientada para o cálculo do vector *x-signature* [Figura de [43]].

$$X[k] = \frac{1}{W} \sum_{d=0}^{W-1} I(u, v) , \text{ com } K=0,1,\dots, l-1 \quad (5.8)$$

## 5.1. PRÉ-PROCESSAMENTO

Na equação 5.8,  $u$  e  $v$ , correspondem respectivamente às equações 5.9 e 5.10, onde  $O(i, j)$  diz respeito ao mapa direccional, calculado anteriormente (ver secção 4.3.1).

$$u = i + \left(d - \frac{W}{2}\right) \cos(O(i, j)) + \left(k - \frac{l}{2}\right) \sin(O(i, j)) \quad (5.9)$$

$$v = j + \left(d - \frac{W}{2}\right) \sin(O(i, j)) + \left(\frac{l}{2} - k\right) \cos(O(i, j)) \quad (5.10)$$

A  $x$ -signature, forma uma onda sinusoidal, com a mesma frequência que as cristas e os vales, da janela considerada, podendo assim, estimar-se a frequência das cristas, a partir desse mesmo vector. Fazendo  $T(i, j)$ , a média dos pixeis, entre dois picos consecutivos no vector  $x$ -signature, então a frequência  $F(i, j)$ , pode ser calculada pela equação 5.11 [53].

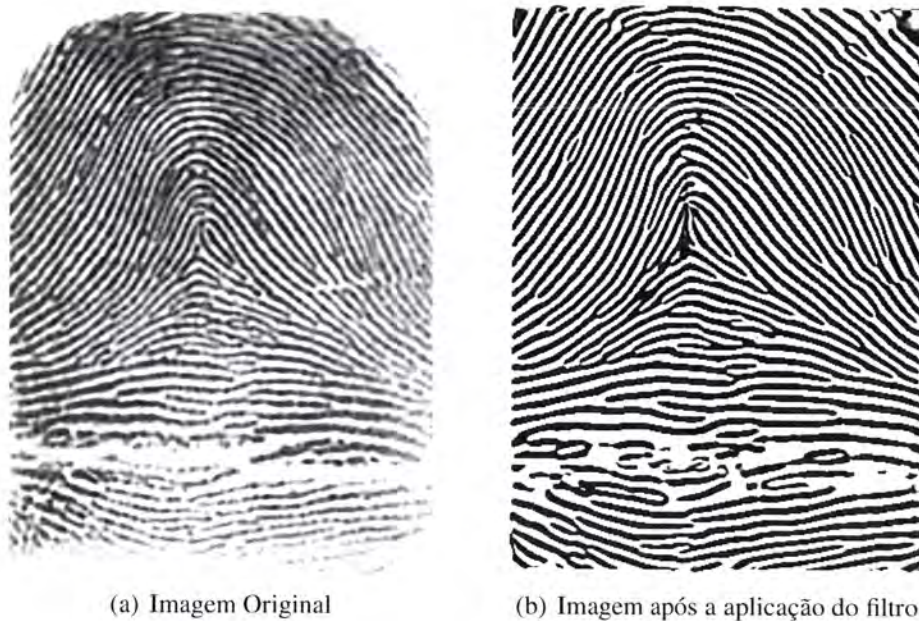
$$F(i, j) = \frac{1}{T(i, j)} \quad (5.11)$$

Se não existirem picos consecutivos no vector  $x$ -signature, então a frequência toma o valor de  $-1$ , de modo a ser diferenciada dos restantes valores.

Para aplicar o filtro de *Gabor* a uma imagem, é necessário especificar 3 parâmetros: a frequência da onda, a orientação do filtro e os desvios padrão do filtro Gaussiano. Os dois primeiros parâmetros correspondem respectivamente, à frequência espacial da imagem e ao mapa direccional. Em relação aos valores dos desvios padrão do filtro Gaussiano, quanto maior forem, maior é a quantidade de ruído que o filtro remove, mas, em contrapartida, são criadas muitas falsas minúcias. Se o valor dos desvios padrão for muito baixo, não existe o problema da criação de falsas minúcias, mas, a redução de ruído é pouco significativa. Em [53], após experiências, chegou-se a um valor de 4.0 para ambos os desvios padrão ( $\sigma_x$  e  $\sigma_y$ ). A optimização da imagem, é feita aplicando a equação 5.12, a cada pixel da imagem.

$$E(i, j) = \sum_{u=-\frac{W_g}{2}}^{\frac{W_g}{2}} \sum_{v=-\frac{W_g}{2}}^{\frac{W_g}{2}} G(u, v : O(i, j), F(i, j) \cdot I(i - u, j - v)) \quad (5.12)$$

Na equação referida,  $I$  representa a imagem original,  $O$  o mapa de orientações,  $F$  o mapa de frequências,  $W_g$  o tamanho da janela do filtro, e finalmente  $E$  a imagem optimizada. A figura 5.10, mostra o resultado da optimização de uma impressão digital, através do filtro de *Gabor*.



**Figura 5.10** Impressão digital otimizada pelo filtro de *Gabor*.

### 5.1.4 Binarização Adaptativa

O processo de binarização, consiste em transformar uma imagem, com diferentes níveis de cinzento, numa outra, que apenas contém pixels de cor branca e preta. Uma vez que a imagem binarizada contém menos informação, torna-se mais fácil, e mais rápido, o seu processamento [54]. O resultado de aplicar este processo, a uma imagem de uma impressão digital, é uma imagem na qual a cor preta, representa as cristas, e a cor branca, os vales [42]. A técnica de binarização (ver equação (5.13)), onde  $L$  representa o valor do limiar, e  $P$  a intensidade do pixel), consiste em percorrer uma imagem, pixel a pixel, e a cada um deles, atribuir o valor 0 ou 1, consoante a intensidade do pixel em questão, estiver acima ou abaixo, de um limiar pré-definido [54].

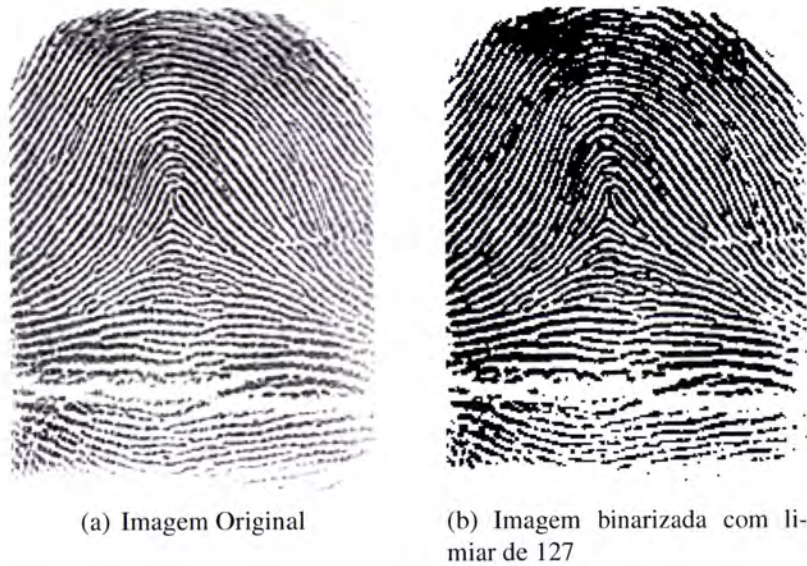
$$f(P) = \begin{cases} 0 & \text{se } P < L \\ 1 & \text{se } P \geq L. \end{cases} \quad (5.13)$$

Usualmente, na ausência de mais informação sobre a imagem, utiliza-se um limiar de 127, que corresponde ao valor central, da escala de cinzento (entre 0 e 255).

A figura 5.11, apresenta o resultado da binarização, de uma impressão digital, utilizando um limiar de 127. Como se pode observar pela imagem, a binarização normal, não tem em consideração, as constantes alterações de intensidade, ao longo da imagem, o que leva ao apa-

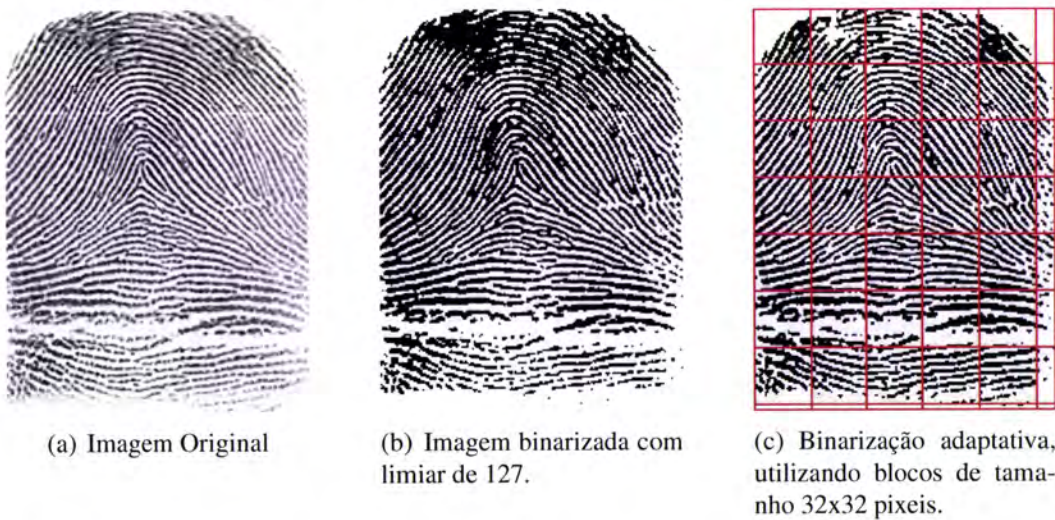
## 5.1. PRÉ-PROCESSAMENTO

recimento de manchas pretas, ou brancas [54]. Para evitar esses problemas, é utilizado um método de binarização adaptativo.



**Figura 5.11** Binarização de uma impressão digital com limiar de 127.

A binarização adaptativa, consiste em dividir a imagem, em blocos  $N \times N$ , e aplicar a cada bloco, uma medida estatística, de modo a definir o limiar para cada bloco [54]. Após o cálculo do limiar, cada pixel do bloco, é então substituído por 0 ou 1, de acordo com a equação (5.13).



**Figura 5.12** Impressão digital binarizada utilizando binarização com limiar fixo e binarização adaptativa.

Através da figura 5.12, é possível comparar os dois tipos de binarização. Como se pode observar, a utilização da binarização adaptativa, origina resultados muito mais próximos da

imagem original.

### 5.1.5 Esqueletização

A última técnica de melhoramento de imagens, usualmente utilizada, antes do processo de extracção de características, é a esqueletização [42, 6].

A esqueletização, tem como objectivo, reduzir a quantidade de pontos de uma imagem, sem afectar a estrutura da imagem original. O esqueleto possui a largura de um pixel, e retêm as características da imagem inicial. A utilização do processo de esqueletização, em impressões digitais, simplifica o processo de extracção de características [55].

Os algoritmos de esqueletização, tendem a ser lentos, uma vez que o varrimento da imagem é feito linha a linha, examinando a vizinhança, e verificando quando um pixel, pode ou não ser apagado [22].

Existem diversos algoritmos de esqueletização, mas aquele que é usualmente mais utilizado, é o algoritmo de *Zhang-Suen*, uma vez que é bastante rápido, e simples de implementar [56].

A ideia base do algoritmo, é verificar se um determinado pixel pode ser eliminado, após a verificação dos seus vizinhos, numa vizinhança de 8 pixeis (figura 5.13).

$P_9$	$P_2$	$P_3$
$P_8$	$P$	$P_4$
$P_7$	$P_6$	$P_5$

**Figura 5.13** Vizinhança de 8 pixeis.

Este algoritmo, é um algoritmo iterativo, que termina quando na última iteração, não tiver sido removido nenhum ponto. Cada iteração do algoritmo, é composta por dois passos. Antes de executar o segundo passo do algoritmo, o pixel que tiver sido marcado para eliminação, é retirado.

O primeiro passo do algoritmo, consiste em verificar as condições descritas em (5.14). No segundo passo, as condições a verificar são idênticas às do primeiro passo, com excepção das condições (c) e (d), que são substituídas por rotações de 180°. Deste modo, o segundo passo remove os pixeis nas posições opostas ao primeiro [57]. As condições do segundo passo do algoritmo são descritas em (5.15).

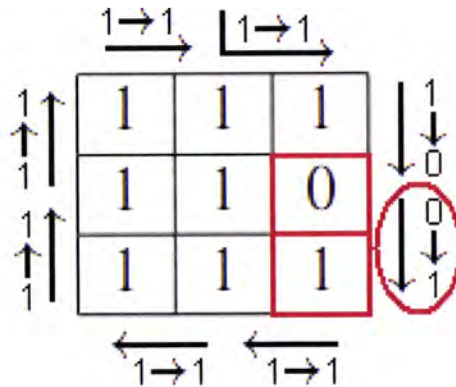
$$\begin{aligned}
 a) \quad & 2 \leq N(p) \leq 6 \\
 b) \quad & C(p) = 1 \\
 c) \quad & p_2 \cdot p_4 \cdot p_6 = 0 \\
 d) \quad & p_4 \cdot p_6 \cdot p_8 = 0
 \end{aligned}
 \tag{5.14}$$

## 5.1. PRÉ-PROCESSAMENTO

$$\begin{aligned} e) \quad & p2.p4.p8 = 0 \\ f) \quad & p2.p6.p8 = 0 \end{aligned} \tag{5.15}$$

Nas condições referidas anteriormente,  $N(p)$ , consiste no número de vizinhos não nulos do pixel central, e é dado pela equação (5.16), e  $C(p)$  diz respeito ao número de conectividade do pixel central, ou seja, o número de transições de branco para preto (0 para 1), iniciando em  $P_2$  e terminando em  $P_9$ . A figura 5.14 representa o cálculo do número de conectividade num exemplo.

$$N(p) = p2 + p3 + p4 + p5 + p6 + p7 + p8 + p9 \tag{5.16}$$



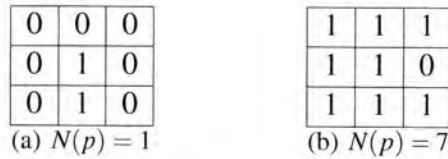
**Figura 5.14** Exemplo do cálculo do número de conectividade ( $C(p) = 1$ ).

As condições que constituem o primeiro passo do algoritmo, são descritas a seguir:

- Condição a)  $2 \leq N(p) \leq 6$

Esta condição indica que o pixel a eliminar, tem de ter mais do que 2 vizinhos pretos, e menos do que 6. O limite inferior de 2, garante que nenhum pixel final, seja eliminado. Um pixel final, é um pixel que apenas tem um vizinho de cor preta. O limite superior de 6, garante que o pixel central a eliminar, pertence à borda do esqueleto, e não ao seu interior. Deste modo, evita-se a remoção de pixels interiores ao corpo do objecto a esqueletizar, prevenindo a criação de linhas separadas. Na figura 5.15 é possível verificar a condição anterior. Em (a), o pixel central é um pixel final, logo não pode ser removido,

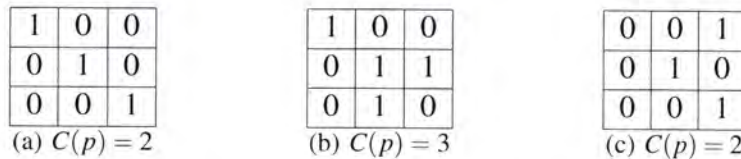
e, nas condições de (b), o pixel central não pode ser removido, uma vez que não pertence às bordas do esqueleto.



**Figura 5.15** Primeira condição do algoritmo de esqueletização.

- Condição b)  $C(p) = 1$

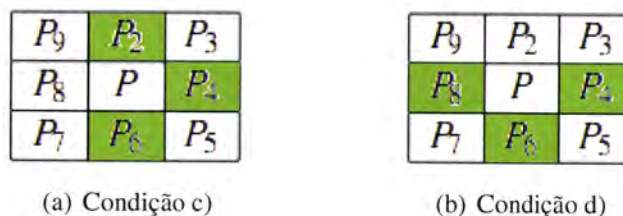
Esta condição, diz que o número de conectividade, do pixel central, tem de ser 1. Desta forma, garante-se que o pixel a remover, não provoca descontinuidades nas linhas do esqueleto. Na figura 5.16, apresentam-se exemplos nos quais o número de conectividade do pixel central, é diferente de 1. Nos referidos exemplos, se o pixel central for removido, cria-se uma descontinuidade entre os restantes pixels, originando falhas nas linhas do esqueleto.



**Figura 5.16** Segunda condição do algoritmo de esqueletização.

- Condição c)  $p_2.p_4.p_6 = 0$  e d)  $p_4.p_6.p_8 = 0$

Estas condições, servem para garantir, que pelo menos um dos pixels de cada equação é 0. A validação destas duas condições, garante que o pixel central pertence à borda do objecto a esqueletizar, não se encontrando assim no interior do corpo do objecto. A figura 5.17 mostra a localização dos pixels aos quais são aplicadas as equações anteriores.



**Figura 5.17** Terceira e quarta condições do algoritmo de esqueletização.

## 5.1. PRÉ-PROCESSAMENTO

No caso de serem verificadas as 4 condições, o pixel central pode ser removido. Uma vez terminado o primeiro passo do algoritmo, é aplicado o segundo passo. As condições do segundo passo são descritas a seguir (as condições (a) e (b) são idênticas nos dois passos):

- Condição e)  $p_2.p_4.p_8 = 0$  e f)  $p_2.p_6.p_8 = 0$

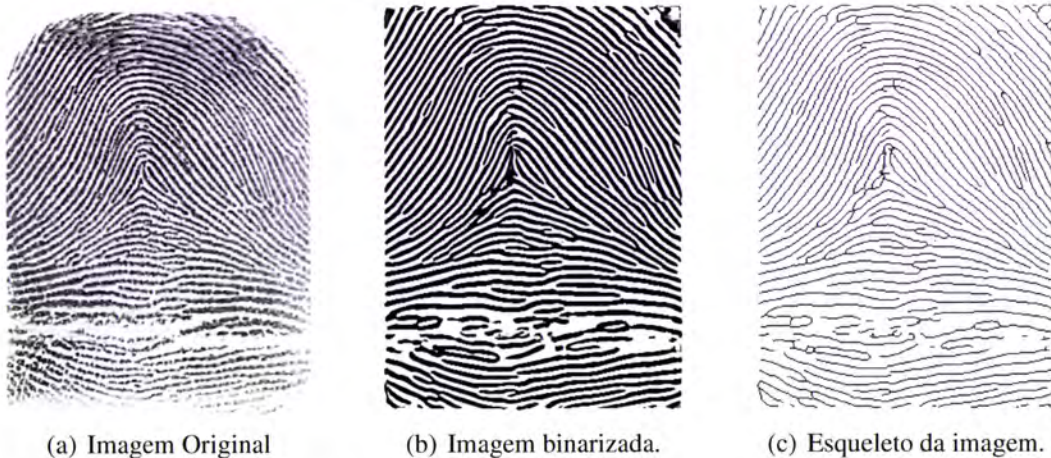
Estas condições, têm o mesmo objectivo das condições (c) e (d) do primeiro passo do algoritmo, ou seja, garantir que o pixel central corresponde a um pixel da borda do esqueleto. Neste passo, essas condições são substituídas por rotações de  $180^\circ$ . A figura 5.18 mostra a localização dos pixels que têm que satisfazer as condições apresentadas.



**Figura 5.18** Condições do segundo passo do algoritmo de esquelização.

Terminado o segundo passo, se não houver elementos marcados para remoção, o algoritmo termina.

A figura 5.19, mostra o resultado de aplicar este algoritmo, a uma imagem de impressão digital.



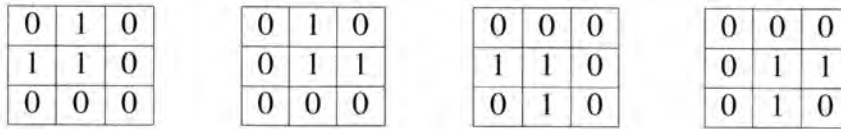
**Figura 5.19** Esquelização de uma impressão digital.

Como se pode observar, o algoritmo cria um esqueleto a partir da imagem binarizada, preservando a sua estrutura inicial.



**5.1.5.1 Limpeza do Esqueleto**

Após a criação do esqueleto da imagem, é necessário proceder a uma limpeza do mesmo, uma vez que o algoritmo de esqueletização, produz um efeito de "escada", ou seja, a presença de degraus na imagem esqueletizada (ver figura 5.20), o que provoca erros no processo de extracção de características [43, 22]. Além deste efeito, podem ainda aparecer, pontos isolados, pontos conectados em forma de *h*, e pontos em que a espessura da linha, não é exactamente de 1 pixel.



**Figura 5.20** Exemplos do efeito escada. Nestas condições, o pixel central deve ser apagado, de forma a não influenciar a extracção de características.

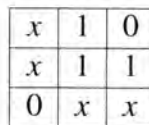
Para remover estes pontos desnecessários, é necessário percorrer a imagem esqueletizada, em janelas 3x3, procurando os padrões, que correspondem a esses mesmos pontos.

No caso dos pontos isolados, conectados em forma de *h*, e naqueles em que a espessura da linha, é maior do que 1 pixel, é feita uma procura na imagem, pelos padrões da figura 5.21. Sempre que é detectado um padrão dessa forma, o pixel central é substituído por 0.



**Figura 5.21** Máscaras para remoção de pontos isolados, *h*-conectados e com espessura maior do que 1.

O padrão que corresponde aos pixels em forma de escada, é visível em 5.22. Neste caso, o pixel central toma o valor de 0, sempre que os pixels marcados com *x* tenham também o valor de 0 [43].



**Figura 5.22** Máscara para remoção do efeito de escada.

Em [22], são propostos, outros tipos de padrões, que podem ser removidos (figura 5.24). Neste caso, tomando como referência a figura 5.23, sempre que os pixels localizados em  $(i, j)$ ,

## 5.2. EXTRACÇÃO DE CARACTERÍSTICAS

---

$(i + 1, j + 1)$  e  $(i - 1, j + 1)$ , tiverem o valor de 1, então, os pixels localizados em  $(i + 1, j)$ ,  $(i, j + 1)$  e  $(i - 1, j)$ , recebem o valor de 0. Deste modo, removem-se os pixels que contornam o objecto, excepto aqueles, que fazem parte do esqueleto do mesmo [22].

$(i - 1, j + 1)$	$(i - 1, j)$	$(i - 1, j + 1)$
$(i, j - 1)$	$(i, j)$	$(i, j + 1)$
$(i + 1, j - 1)$	$(i + 1, j)$	$(i + 1, j + 1)$

**Figura 5.23** Localização do pixel central e seus vizinhos.

$x$	0	1
$x$	1	0
$x$	0	1

**Figura 5.24** Máscara para remoção dos pixels que contornam o objecto.

Além de ser feita a pesquisa, para todas as máscaras aqui incluídas, é ainda necessário, pesquisar na imagem, por todas as rotações possíveis, de cada máscara. O conjunto total das máscaras a utilizar, pode ser consultado no apêndice A.

## 5.2 Extracção de Características

Uma vez feita a optimização da imagem, através dos métodos anteriores, é agora possível, com maior exactidão, proceder à extracção dos aspectos característicos da mesma. O cálculo do ponto de referência é um passo crucial, uma vez que, é através dele que se consegue uma invariância à rotação e ao deslocamento, da impressão digital [38]. O ponto de referência de uma impressão digital está localizado nas linhas que apresentam a maior curvatura côncava. Na maioria dos casos, estas linhas representam a localização do núcleo da impressão digital, podendo este ser utilizado como ponto de referência da mesma. As minúcias extraídas da imagem, juntamente com o ponto de referência, são utilizadas, posteriormente, no algoritmo de identificação, de modo a fazer corresponder uma dada impressão digital, a uma outra, numa base de dados.

### 5.2.1 Cálculo do ponto de Referência

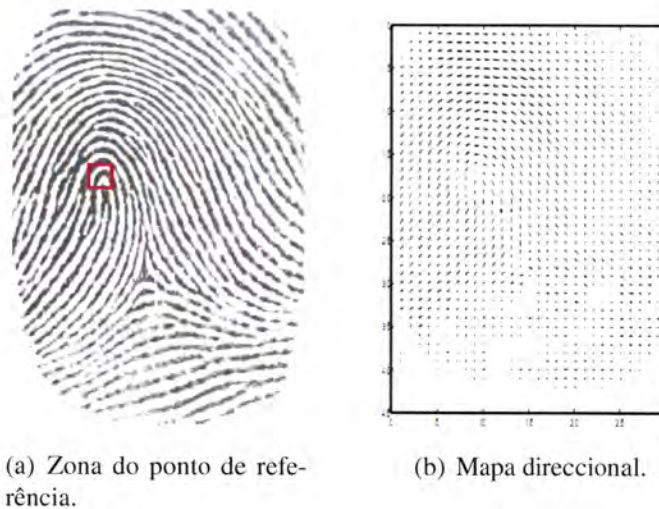
A consistência no cálculo da posição do ponto de referência numa impressão digital é crucial, uma vez que este deve ser único, e não deve haver ambiguidade na sua detecção [38]. A

## CAPÍTULO 5. TÉCNICAS PARA O PROCESSAMENTO DE IMPRESSÕES DIGITAIS

localização do ponto de referência em imagens da mesma impressão digital, deve ter sempre a mesma posição relativa na imagem. Se tais factores não se verificarem, a extracção dos pontos característicos, aliados à posição do ponto de referência, resulta em diferentes resultados, criando ambiguidades no momento de comparação das impressões digitais [38].

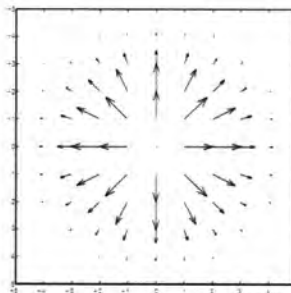
Neste trabalho, o método utilizado para o cálculo do ponto de referência, é descrito em detalhe em [58, 59, 60, 61], e consiste na aplicação de um filtro complexo à imagem da impressão digital. Este filtro funciona no domínio da frequência, através da utilização da transformada de Fourier. A sua construção é baseada na teoria dos números complexos, originando assim o nome de filtro complexo.

A aplicação do filtro, permite extrair a localização do núcleo da impressão digital, que será utilizado como ponto de referência [59].



**Figura 5.25** Impressão digital e respectivo mapa direccional. [Figuras de [58]]

Na figura 5.26 é possível ver uma aproximação gráfica do filtro utilizado para a extracção do ponto de referência.



**Figura 5.26** Representação gráfica do filtro complexo para a extracção do ponto de referência.[Figura de [58]]

## 5.2. EXTRACÇÃO DE CARACTERÍSTICAS

---

A fórmula do filtro complexo é dada pela equação (5.17), onde  $x$  e  $y$  representam coordenadas,  $i$  representa o número imaginário e,  $\sigma^2$ , é um parâmetro de afinamento da janela gaussiana.

$$h(x,y) = (x + iy)e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (5.17)$$

O algoritmo consiste em aplicar o filtro (5.17) a blocos do mapa direccional, com o mesmo tamanho, através de uma convolução dos dois. A figura 5.27 mostra o resultado da aplicação do filtro à impressão digital em 5.25.



**Figura 5.27** Resposta do filtro complexo aplicado a uma impressão digital. [Figura de [58]]

A implementação do cálculo do ponto de referência consiste numa convolução entre o filtro complexo (fórmula em (5.17)) e o mapa direccional, da impressão digital em questão [59].

O teorema da convolução, diz que a convolução de uma imagem  $h(x,y)$  com uma outra imagem  $z(x,y)$ , origina uma terceira imagem  $I(x,y)$ , estando os valores de  $x$  e  $y$  compreendidos no intervalo  $[0, M - 1]$  e  $[0, N - 1]$ , respectivamente, sendo  $M$  e  $N$  a quantidade de linhas e colunas dessas imagens. Nestas condições, são então verdadeiras as equações (5.18) e (5.19), onde  $*$  representa o operador de convolução [30].

$$I(x,y) = h(x,y) * z(x,y) \quad (5.18)$$

$$I(x,y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} h(m,n)z(x-m,y-n) \quad (5.19)$$

## CAPÍTULO 5. TÉCNICAS PARA O PROCESSAMENTO DE IMPRESSÕES DIGITAIS

---

Tendo em  $I(x,y)$ , a aplicação do filtro ao mapa direccional, procura-se o ponto máximo de magnitude  $(x,y)$ . Este ponto será então utilizado como o ponto de referência da impressão digital em questão.

A implementação do algoritmo proposto, para o cálculo do ponto de referência, segue um conjunto de passos [62, 63], que são descritos a seguir:

1. Inicialização de dois vectores,  $V_x$  e  $V_y$ , com os valores pertencentes ao intervalo fechado  $[-16, 16]$ .
2. Criação de uma máscara ( $M$ ) de zeros, sob a forma de matriz, cujo tamanho é igual ao tamanho dos vectores anteriores (33x33).
3. Recorrendo aos vectores anteriores, constrói-se o filtro complexo ( $F_c$ ), recorrendo à aplicação da equação (5.17). Na referida equação, os elementos  $x$  e  $y$  correspondem a cada posição dos vectores  $V_x$  e  $V_y$ ,  $i$  corresponde ao número imaginário puro e para  $\sigma^2$ , é utilizado o valor de 55. Cada posição da matriz  $M$  é substituída pelo cálculo desta equação para cada ponto dos vectores.
4. Seja  $O$  o mapa direccional correspondente à impressão digital, para a qual se pretende calcular o ponto de referência. O mapa direccional  $O$ , é dividido em blocos ( $B$ ), não sobrepostos, de tamanho 33x33.
5. Aplica-se a transformada de Fourier ao filtro  $F_c$ , originando  $F_c f$ .
6. A cada bloco  $B$ , calculado anteriormente, é também aplicada a transformada de Fourier, originando  $B f$ .
7. Faz-se a convolução de  $F_c f$  com cada  $B f$ . No espaço da frequência, a convolução de duas imagens, corresponde a fazer uma multiplicação ponto a ponto das matrizes das duas imagens.
8. Processados todos os blocos, reconstrói-se a imagem, posicionando cada bloco no seu lugar original.
9. O ponto de resposta máxima do filtro, corresponde ao máximo da matriz da imagem. Assim, calculando a posição do valor máximo da matriz da imagem, obtém-se a posição do ponto de referência, e respectivas coordenadas  $(x,y)$ .

A figura 5.28 representa uma impressão digital, com a localização do respectivo ponto de referência.

## 5.2. EXTRACÇÃO DE CARACTERÍSTICAS

---



**Figura 5.28** Localização do ponto de referência, numa impressão digital.

### 5.2.2 Extracção de Minúcias

Os métodos propostos nas secções anteriores, produzem uma imagem otimizada, e com linhas cuja largura é apenas um pixel. Sobre esta imagem é agora possível, de forma bastante simples, aplicar algoritmos que extraem os pontos característicos. Uma das formas mais simples de fazer a extracção de minúcias, é percorrer toda a imagem, verificando a relação que cada pixel tem com os seus vizinhos [43].

Um dos métodos mais comuns para a extracção de minúcias, em imagens esqueletizadas, é o método do número de cruzamentos (*Crossing Number - CN*) [42, 64].

Este método permite determinar se um pixel corresponde a uma minúcia, através do somatório de transições de preto para branco, que existem num conjunto de 9 pixels (ou seja, uma janela de tamanho 3x3) [43].

$P_0$	$P_1$	$P_2$
$P_7$	$P$	$P_3$
$P_6$	$P_5$	$P_4$

**Figura 5.29** Vizinhança de 8 pixels, para o cálculo do número de cruzamentos.

O cálculo do CN, consiste em dividir a imagem esqueletizada, em janelas de tamanho 3x3, e aplicar, a cada uma dessas janelas, de acordo com a figura 5.29, a fórmula (5.20) [64, 43, 65].

$$Cn(P) = \frac{1}{2} \sum_{i=1 \dots 8} |val(P_{mod(i,8)}) - val(P_{i-1})| \quad (5.20)$$

Na fórmula (5.20),  $Cn(P)$  representa o cálculo do algoritmo para o pixel  $P$  e  $val(P_i)$  tem o valor de 1 caso o pixel  $P_i$  seja preto e 0 caso contrário. O cálculo é feito conforme a disposição

## CAPÍTULO 5. TÉCNICAS PARA O PROCESSAMENTO DE IMPRESSÕES DIGITAIS

ilustrada em 5.29, e a expressão  $mod(i, 8)$ , é a função que calcula o resto da divisão inteira de  $i$  por 8.

O tipo de cada pixel, depende do valor calculado pelo CN. Dependendo do tipo de relação que tem com os seus vizinhos, o pixel em questão será uma minúcia ou uma linha simples. A figura 5.30 mostra a relação entre o valor do CN e o tipo de minúcia correspondente.

Valor do Cn	Tipo de Minúcia
0	Ponto Isolado
1	Terminação
2	Crista Normal
3	Bifurcação
4 ou mais	Não Considerado

**Figura 5.30** Propriedades do algoritmo número de cruzamentos.

O método proposto apenas localiza minúcias do tipo terminação ou bifurcação, uma vez que, como foi dito anteriormente (ver capítulo 2.1.2), todas as outras podem ser obtidas por combinação destas duas. A figura 5.31 apresenta exemplos que podem ocorrer na pesquisa de minúcias na imagem.

<table border="1" style="border-collapse: collapse;"> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td></tr> </table>	1	0	0	0	1	0	0	0	1	<table border="1" style="border-collapse: collapse;"> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </table>	1	0	0	0	1	0	0	0	0	<table border="1" style="border-collapse: collapse;"> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> </table>	1	0	1	0	1	0	1	0	0
1	0	0																											
0	1	0																											
0	0	1																											
1	0	0																											
0	1	0																											
0	0	0																											
1	0	1																											
0	1	0																											
1	0	0																											
(a) Crista normal (Cn=2).	(b) Terminação (Cn=1).	(c) Bifurcação (Cn=3).																											

**Figura 5.31** Exemplo de padrões no cálculo do número de cruzamentos.

À medida que as minúcias vão sendo detectadas pelo algoritmo, são guardadas numa estrutura, de acordo com 5.32. Para cada minúcia, são guardadas as coordenadas espaciais da mesma na imagem (representadas por  $(x,y)$ ) e o tipo, que representa o tipo de minúcia extraída: terminação ou bifurcação.

$$M = [[(x,y), Tipo], \dots] \tag{5.21}$$

**Figura 5.32** Estrutura das minúcias extraídas pelo algoritmo CN.

## 5.3. PÓS-PROCESSAMENTO

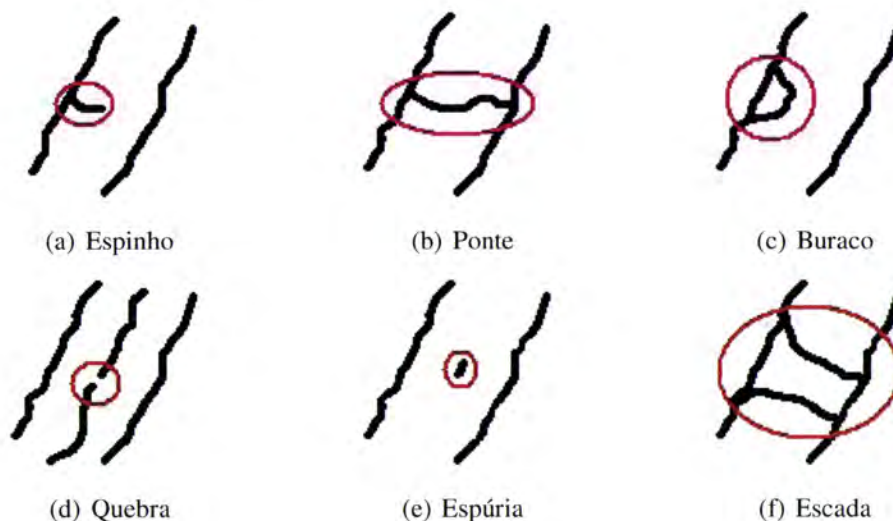
### 5.3 Pós-processamento

A última etapa da construção de um sistema AFIS, consiste no pós-processamento. Após a extracção de características, é necessário proceder a uma validação das minúcias encontradas, uma vez que os métodos utilizados para o melhoramento da imagem, podem originar falsas minúcias. Após feita a validação das minúcias, e recorrendo ao ponto de referência calculado anteriormente (ver 5.2.1), é então, construída uma representação da impressão digital, que poderá ser posteriormente adicionada a uma base de dados, para futura comparação.

#### 5.3.1 Validação de Minúcias

O pré-processamento tem como objectivo reduzir o número de degradações presentes numa imagem, no entanto, também pode ser responsável pela introdução de erros na mesma. Esta introdução de erros na imagem, podem ser falsamente detectados como minúcias, o que pode induzir o sistema AFIS em erro. A etapa de validação de minúcias tem como objectivo, a localização e eliminação de pontos, que foram, incorrectamente, marcados como minúcias [43].

O algoritmo proposto para a validação das minúcias funciona sobre duas listas separadas, calculadas pelo CN: uma lista de minúcias do tipo terminações, e uma outra, do tipo bifurcações. A análise é feita em janelas de tamanho  $W \times W$ , e, para cada ponto marcado como minúcia, o algoritmo decide se é uma minúcia verdadeira ou falsa. A figura 5.33 mostra exemplos de falsas minúcias, que podem ocorrer numa imagem.



**Figura 5.33** Exemplos de falsas minúcias.



## CAPÍTULO 5. TÉCNICAS PARA O PROCESSAMENTO DE IMPRESSÕES DIGITAIS

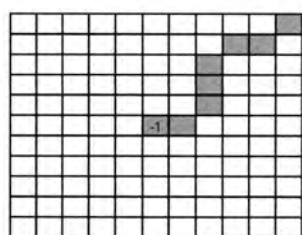
---

Os passos necessários para a validação de uma minúcia, pelo algoritmo proposto, são descritos a seguir [43, 42].

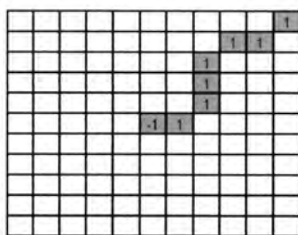
Para um ponto marcado como terminação, pelo algoritmo CN, a sua validação é feita da seguinte maneira:

1. Cria-se uma máscara  $M$  de tamanho  $W \times W$ , retirada da imagem esqueletizada, e centrada no pixel correspondente à minúcia em questão.
2. Marcar com  $-1$  o pixel central de  $M$ . Este é o pixel que correspondem à minúcia candidata a validação.
3. Marcar com  $1$  todos os pixels da máscara  $M$  que de alguma forma têm ligação com o pixel central.
4. Fazer uma volta completa, pelas margens da máscara, verificando o valor dos pixels conectados ao pixel central. Deste modo, calcula-se  $T_{01}$ , que corresponde ao número de transições de  $0$  para  $1$ , ou seja, o número total de  $1$  nas margens da máscara, que se encontram conectados ao pixel central.
5. Caso  $T_{01} = 1$ , então a minúcia é marcada como sendo uma terminação verdadeira.

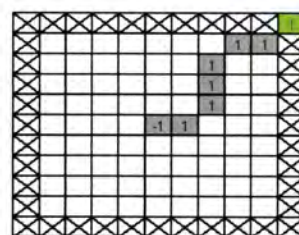
Nas figuras 5.34 e 5.35, apresentam-se exemplos da aplicação do algoritmo, passo-a-passo, utilizando uma janela de tamanho  $11 \times 11$ . A primeira figura representa a aplicação do algoritmo a uma minúcia válida, e a segunda, a uma minúcia falsa.



(a) Máscara centrada na minúcia.



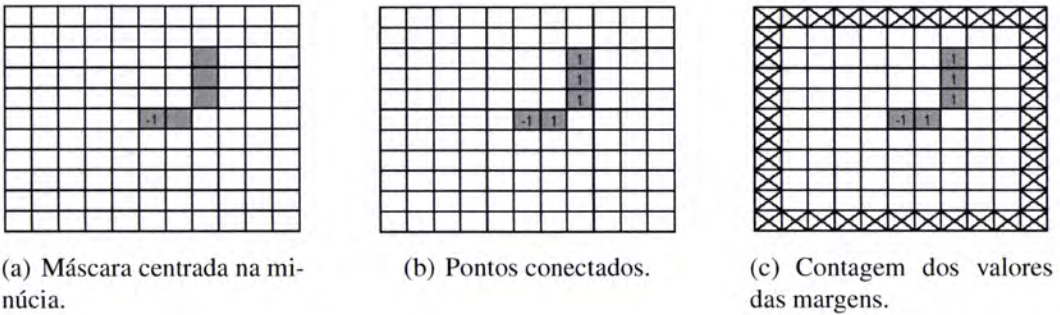
(b) Pontos conectados.



(c) Contagem dos valores das margens.

**Figura 5.34** Validação de uma minúcia do tipo terminação.

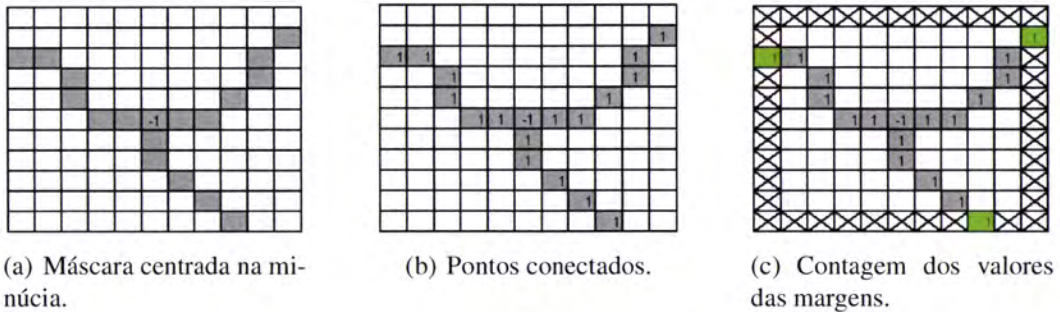
### 5.3. PÓS-PROCESSAMENTO



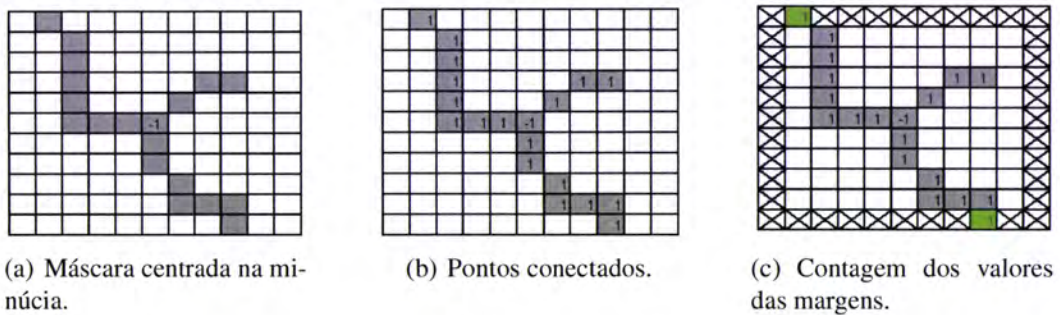
**Figura 5.35** Rejeição de uma minúcia do tipo terminação.

O processo de validação de uma minúcia do tipo bifurcação é semelhante ao anterior. No caso das bifurcações, a minúcia, apenas é marcada como sendo verdadeira, caso  $T_{01} = 3$  [43, 42].

As figuras 5.36 e 5.37 mostram, respectivamente, a validação e rejeição, de uma minúcia do tipo bifurcação, pelo algoritmo.



**Figura 5.36** Validação de uma minúcia do tipo bifurcação.



**Figura 5.37** Rejeição de uma minúcia do tipo bifurcação.

A aplicação deste algoritmo, às listas de minúcias detectadas no processo de extracção de minúcias, provoca uma grande redução do número de minúcias existentes na imagem, como se pode verificar nas figuras 5.38 e 5.39.

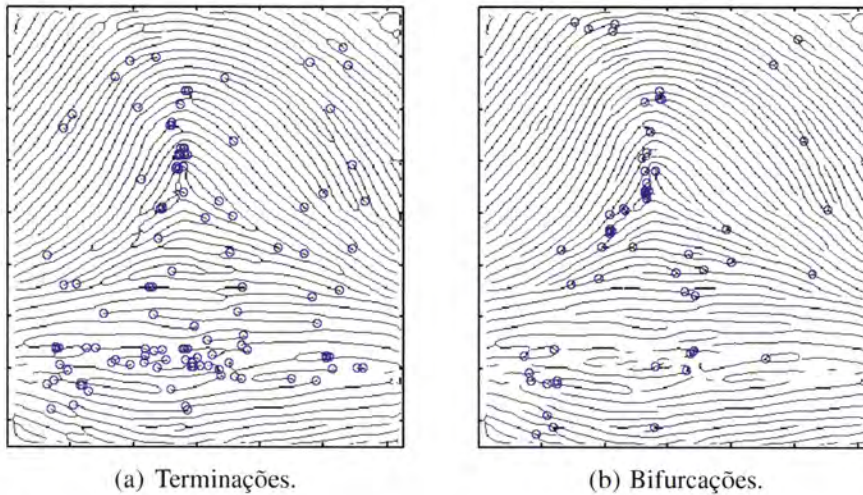


Figura 5.38 Minúcias extraídas pelo algoritmo CN, antes da validação.

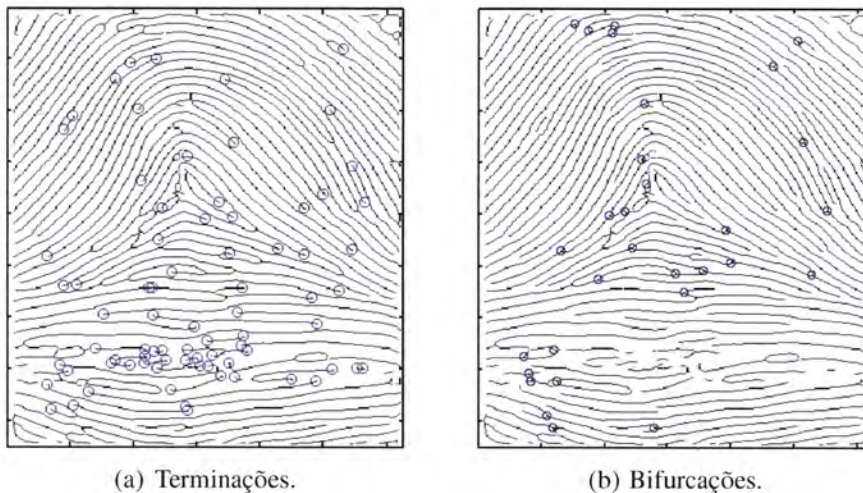


Figura 5.39 Minúcias consideradas válidas, após o algoritmo de validação.

As referidas figuras, apresentam marcadas com círculos azuis, as minúcias (terminações e bifurcações) extraídas pelo algoritmo CN, e as mesmas minúcias, após a aplicação do algoritmo de validação.

### 5.3.2 Representação da Impressão Digital

O último passo necessário, antes de se inserir a impressão digital numa base de dados, é criar uma representação da mesma, que relacione a posição das minúcias, em relação ao ponto de referência. Esta representação consiste em calcular, para cada uma das minúcias encontradas na imagem, a sua distância ao ponto de referência, e o ângulo que fazem com o mesmo.

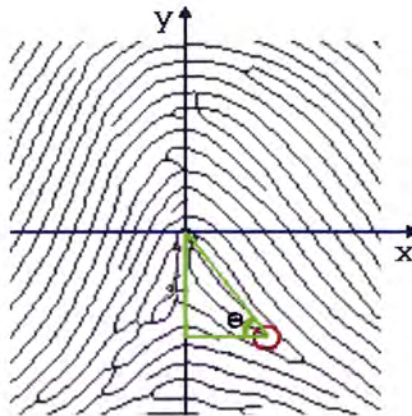
O cálculo da distância de cada minúcia ao ponto de referência, é feito recorrendo às coor-

### 5.3. PÓS-PROCESSAMENTO

denadas espaciais dos dois pontos. A cada uma das minúcias, é aplicada a equação (5.22), que obtém assim, a distância euclidiana entre os dois pontos [66, 38]. Fazem parte desta equação, as coordenadas espaciais da minúcia (representadas por  $M_x$  e  $M_y$ ), e as coordenadas espaciais do ponto de referência (representadas por  $PR_x$  e  $PR_y$ ).

$$D = \sqrt{(PR_x - M_x)^2 + (PR_y - M_y)^2} \quad (5.22)$$

O cálculo do ângulo que cada minúcia faz com o ponto de referência, é feito através da tangente. O ponto de referência pode ser utilizado como sendo a origem de um sistema de eixos (ver figura 5.40) [64, 38]. Assim, o ângulo que cada minúcia faz com o ponto de referência, pode ser calculado através da equação 5.23.



**Figura 5.40** Ponto de referência como origem de um sistema de eixos para o cálculo do ângulo das minúcias.

$$\theta = \tan^{-1} \left( \frac{M_y - PR_y}{M_x - PR_x} \right) \quad (5.23)$$

Uma vez calculado, para todas as minúcias detectadas na imagem, as suas distâncias e ângulos, em relação ao ponto de referência, é possível construir uma representação, em coordenadas polares ( $D, \theta$ ), para armazenar as impressões digitais numa base de dados. A figura 5.41 mostra um exemplo de uma representação possível, onde  $D$ , representa a distância ao ponto de referência,  $\theta$ , o ângulo que a minúcia faz com o mesmo,  $k$ , o número total de minúcias, e  $T$ , o tipo de minúcia: terminação ou bifurcação.

$$[[D, \theta, T]_1, [D, \theta, T]_2, \dots, [D, \theta, T]_k] \quad (5.24)$$

**Figura 5.41** Representação de uma impressão digital, na forma de uma lista.

Finalmente, recorrendo à classe da impressão digital, previamente calculada, é possível guardar a representação da impressão digital, numa base de dados, para futura comparação.

## 5.4 Origem do código utilizado

O código utilizado na implementação dos algoritmos propostos neste capítulo, teve diferentes origens.

A aplicação ImageJ, desenvolvida inteiramente em JAVA, foi utilizada para fazer a leitura e escrita das imagens. Foi também utilizado o algoritmo de normalização de contraste, assim como os filtros de mediana e esqueletização, implementados nesta ferramenta.

O filtro de Gabor, foi desenvolvido na linguagem de programação C, e faz parte da biblioteca de funções biométricas, *FVS*. A biblioteca foi desenvolvida, na sua totalidade, na linguagem de programação C, utilizando a aplicação *ImageMagick*, como biblioteca, para ler e escrever imagens.

A algoritmo do cálculo do ponto de referência, foi implementado por Luiji Rosa, recorrendo ao *Octave*. A implementação faz uso da transformada de Fourier, implementada no *Octave*.

Todos os restantes algoritmos (*root filtering*, binarização, limpeza de esqueleto, *crossing number* e validação de minúcias), foram desenvolvidos no âmbito deste trabalho. Todas as implementações foram feitas recorrendo à linguagem de programação JAVA, com excepção do algoritmo *root filtering*, que foi implementado recorrendo ao *Octave*, e às funções da transformada de Fourier e transformada inversa de Fourier, implementadas na aplicação.

## Capítulo 6

# Identificação com impressões digitais

No final das fases de classificação e de processamento da imagem, da impressão digital, esta, pode ser armazenada numa base de dados. De modo a ser possível fazer corresponder uma dada impressão digital a uma outra arquivada na base de dados, é necessário utilizar um algoritmo que faça a comparação dos pontos característicos da impressão digital a comparar, com os mesmos pontos, das impressões que se encontram arquivadas na base de dados.

Em [38, 43, 66, 22], no que diz respeito à fase de comparação das impressões digitais, apenas é indicado, que é necessário fazer um alinhamento da impressão digital a comparar, com as impressões digitais arquivadas, comparando depois as minúcias sobrepostas e o seu tipo.

No entanto, não há nenhuma descrição de algoritmos, ou métodos, utilizados para fazer este alinhamento e posterior comparação das sobreposições das minúcias. Por este motivo, foi desenvolvido um algoritmo, que, utilizando a representação da impressão digital, proposta em 5.3.2, consiga fazer corresponder uma impressão digital, a uma outra, arquivada numa base de dados.

### 6.1 Algoritmo de comparação proposto

O processo de identificação proposto, consiste, em primeiro lugar, na classificação da impressão digital, a comparar, de modo a seleccionar, na base de dados, o conjunto de impressões, com as quais vai ser comparada. Após o processo de classificação (ver capítulo 4), a impressão digital, é sujeita aos processos de pré-processamento e extracção de características. Este processamento é idêntico ao aplicado às impressões digitais que constam na base de dados (processos descritos em detalhe no capítulo 5). No final destes processos, a impressão digital a comparar, encontra-se agora, na mesma representação que as impressões digitais que constam

## CAPÍTULO 6. IDENTIFICAÇÃO COM IMPRESSÕES DIGITAIS

na base de dados. É agora possível, comparar a impressão digital com a base de dados, utilizando o algoritmo proposto a seguir (no apêndice B, é fornecido um diagrama com todos os passos do algoritmo).

Seja  $I$  a impressão digital a comparar e  $BD$  a base de dados das impressões digitais. Após a impressão digital ser classificada, temos  $C$ , que corresponde à classe da mesma. Fazendo uma pesquisa em  $BD$ , por todas as impressões digitais da classe  $C$ , cria-se assim uma lista  $L$ , de todas as impressões digitais que é necessário comparar com  $I$ , de modo a tentar obter uma correspondência.

O fluxograma 6.1, ilustra os passos do algoritmo de comparação, proposto nesta dissertação. Os algoritmos 6.1 e 6.2, apresentam, em pseudo-código, o funcionamento dos blocos (1) e (2), marcados no fluxograma 6.1.

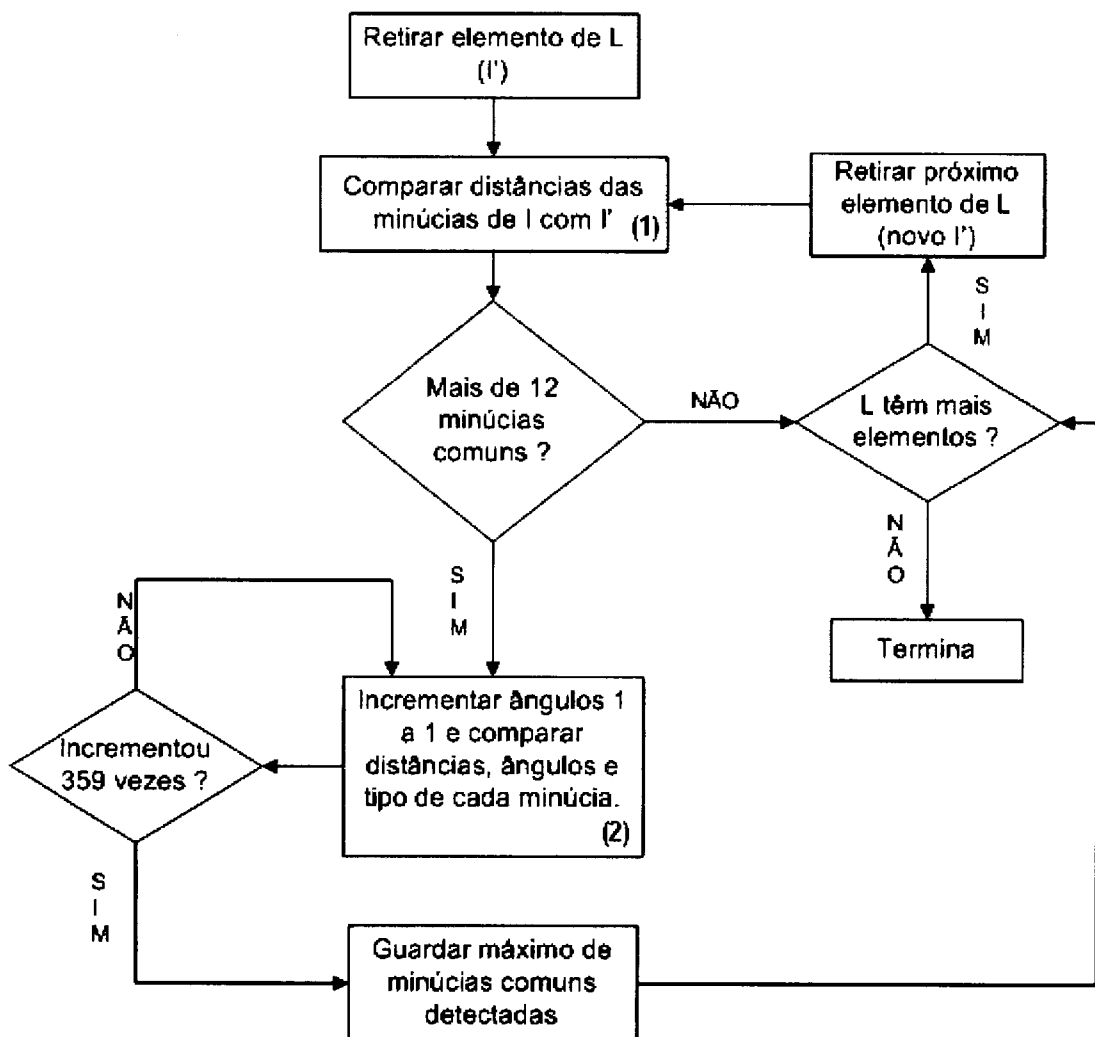


Figura 6.1 Fluxograma do algoritmo de comparação.

## 6.1. ALGORITMO DE COMPARAÇÃO PROPOSTO

---

Para cada impressão digital presente na lista calculada anteriormente ( $L$ ), é retirado uma imagem de cada vez ( $I'$ ). Para cada uma das imagens, são comparadas as distâncias das suas minúcias, com as distâncias das minúcias da impressão digital a comparar ( $I$ ). O algoritmo 6.1, representa este passo em pseudo-código, que corresponde ao bloco (1) do fluxograma 6.1. No referido algoritmo,  $idcmp$  e  $idbd$  correspondem, respectivamente, à representação da impressão digital a comparar ( $I$ ), e à representação da impressão digital de  $L$  actual. O valor do erro máximo considerado, é dado por  $err$ .

```
FUNCTION CalculaDistancias(idcmp, idbd, err)
  VARS STRUCTURE:
    idcmp, idbd ← [(Dst, Ang, Type), ...]
  START:
    mComuns ← 0
    for i ← 0 to idcmp.numberOfElements {
      for j ← 0 to idbd.numberOfElements {
        if (idbd[j].DST-err > idcmp[i].DST < idbd[j].DST+err)
          {
            mComuns++
            idcmp.removeElementAt(j)
          }
      }
    }
  END:
  return mComuns
END FUNCTION
```

**Algoritmo 6.1** Pseudo-código para o cálculo das minúcias comuns entre duas representações.

Caso o número de minúcias, cujas distâncias sejam idênticas (considerando uma margem de erro  $\epsilon$ ), seja menor que 12 (em 2.1.2, foi visto que são necessários pelo menos 12 pontos característicos comuns para que duas impressões digitais sejam consideradas iguais), então essa impressão digital é descartada, uma vez que não tem pontos suficientes em comum com a imagem a comparar ( $I$ ).

Caso o número de minúcias comuns entre as duas impressões digitais seja igual, ou superior a 12, é necessário comparar os seus ângulos e tipo de minúcia.

De forma a resolver o problema da rotação das impressões digitais, são efectuados diversos incrementos de  $1^\circ$ , de cada vez (até um máximo de  $359^\circ$ , que corresponde a uma rotação total



## CAPÍTULO 6. IDENTIFICAÇÃO COM IMPRESSÕES DIGITAIS

da impressão digital), a todos os ângulos das minúcias de I.

```
FUNCTION ComparaMinucias(idcmp, idbd, ed, ea)
  VARS STRUCTURE:
    idcmp, idbd ← [(Dst,Ang,Type),...]
  START:
    minuciasComuns, maxMinucias ← 0
    for i ← 0 to 359 {
      for k ← 0 to idcmp.numberofElements {
        idcmp[k].Ang = idcmp[k].Ang+1
      }
      for l ← 0 to idcmp.numberofElements {
        for j ← 0 to idbd.numberofElements {
          if (idbd[j].DST-ed > idcmp[l].DST < idbd[j].DST+ed) {
            if (idbd[j].ANG-ea > idcmp[l].ANG < idbd[j].ANG+ea) {
              if (idbd[j].TYPE == idcmp[l].TYPE) {
                minuciasComuns++;
                idcmp.removeElementAt(j)
              }
            }
          }
        }
      }
      if (minuciasComuns > maxMinucias) {
        maxMinucias = minuciasComuns
      }
    }
  END:
  return maxMinucias
END FUNCTION
```

**Algoritmo 6.2** Pseudo-código para a comparação das minúcias.

Em cada um dos incrementos, são comparadas as distâncias, ângulos e tipo de minúcia, de todas as minúcias da impressão digital a comparar (*I*), com a impressão digital em questão (*I'*), guardando-se o valor máximo de minúcias comuns, entre as duas. Este processo, corresponde ao bloco (2) do fluxograma 6.1, e é apresentado em pseudo-código, no algoritmo 6.2. No

## 6.1. ALGORITMO DE COMPARAÇÃO PROPOSTO

---

referido algoritmo, *idcmp* e *idbd* correspondem, respectivamente, à representação da impressão digital a comparar ( $I$ ), e à representação da impressão digital de  $L$ , actual. Em relação aos valores de erro máximos, tanto para a distância, como para o ângulo, são dados por  $ed$  e  $ea$ .

Terminada a rotação da impressão digital, é guardado, numa lista ordenada ( $M$ ), o valor máximo de minúcias em comum, entre as duas impressões digitais, juntamente com a identificação da impressão digital em questão ( $I'$ ). Procede-se desta forma, para todas as impressões digitais que pertencem à lista  $L$ .

Terminadas as impressões digitais na lista  $L$ , a lista ordenada  $M$ , contém as impressões digitais em que foram detectados mais pontos característicos em comum com a impressão digital a comparar ( $I$ ).

Resumindo, o funcionamento do algoritmo proposto, é dado pelos seguintes passos:

1. Classificar a impressão digital a comparar.
2. Submeter a impressão digital a comparar, aos processos de pré-processamento e extração de características.
3. Extrair para uma lista, todas as impressões digitais da base de dados, cuja classe corresponda à classe da impressão digital a comparar.
4. Para cada impressão digital, da lista anterior, comparar a distância das suas minúcias, com a distância das minúcias da impressão digital a comparar. Caso o número de minúcias, cujas distâncias iguais, seja inferior a 12, a impressão digital da lista é descartada.
5. Incrementar, 1º de cada vez, os ângulos das minúcias da impressão digital a comparar. A cada incremento, são comparados os ângulos, distâncias, e tipo de minúcia, com a impressão digital da lista. O valor máximo de minúcias comuns detectadas, entre as duas impressões digitais, é guardado numa lista.
6. Terminadas as impressões digitais da lista, a lista ordenada, contém as impressões digitais que apresentam maiores semelhanças, com a impressão digital a comparar.

## **6.2 Exemplo de aplicação do algoritmo proposto**

De seguida, apresenta-se um pequeno exemplo, da aplicação deste algoritmo. Para este exemplo, foram criadas 4 representações de impressões digitais, uma para comparação, e as restantes 3 para a criação de uma base de dados. As impressões digitais que pertencem à base de dados, apresentam as seguintes características:

- Existe uma impressão digital que têm menos de 13 minúcias em comum com as outras duas.
- Uma das impressões da base de dados, corresponde à impressão a comparar, com uma rotação de 5°.
- A impressão digital que resta, tem suficientes minúcias em comum para ser comparada, mas é diferente das outras duas.

A figura 6.2 representa uma possível impressão digital ( $F_C$ ) que se pretende comparar com uma base de dados, já classificada, processada e na representação proposta. A figura 6.3 representa uma base de dados, composta por 3 impressões digitais ( $FBD_1, FBD_2$  e  $FBD_3$ ), pertencentes à mesma classe que  $F_C$ .

```
[
[3.2,70,Terminação],    [18.1,41,Bifurcação],    [28.9,67,Terminação],    [36.2,2,Terminação],
[54.4,131,Terminação], [57.7,23,Bifurcação],    [60.8,88,Bifurcação],    [71.2,107,Bifurcação],
[83.4,140,Terminação], [90.9,31,Terminação],    [101.7,178,Bifurcação],  [112.3,216,Terminação],
[118.5,307,Bifurcação], [123.1,25,Terminação],   [131.3,44,Bifurcação],   [149.5,73,Terminação],
[156.9,22,Bifurcação], [175.2,14,Terminação],   [182.6,129,Terminação],  [198.3,19,Bifurcação],
[223.5,178,Terminação], [237.8,12,Bifurcação],   [243.3,169,Terminação],  [286.6,186,Terminação],
[291.1,27,Terminação]
]
```

**Figura 6.2** Exemplo de uma representação de uma impressão digital.

Aplicando o algoritmo de comparação proposto, verifica-se que, após a contagem das minúcias que se encontram à mesma distância,  $FBD_1$  tem 0 minúcias à mesma distância,  $FBD_2$ ,

## Capítulo 6

# Identificação com impressões digitais

No final das fases de classificação e de processamento da imagem, da impressão digital, esta, pode ser armazenada numa base de dados. De modo a ser possível fazer corresponder uma dada impressão digital a uma outra arquivada na base de dados, é necessário utilizar um algoritmo que faça a comparação dos pontos característicos da impressão digital a comparar, com os mesmos pontos, das impressões que se encontram arquivadas na base de dados.

Em [38, 43, 66, 22], no que diz respeito à fase de comparação das impressões digitais, apenas é indicado, que é necessário fazer um alinhamento da impressão digital a comparar, com as impressões digitais arquivadas, comparando depois as minúcias sobrepostas e o seu tipo.

No entanto, não há nenhuma descrição de algoritmos, ou métodos, utilizados para fazer este alinhamento e posterior comparação das sobreposições das minúcias. Por este motivo, foi desenvolvido um algoritmo, que, utilizando a representação da impressão digital, proposta em 5.3.2, consiga fazer corresponder uma impressão digital, a uma outra, arquivada numa base de dados.

### 6.1 Algoritmo de comparação proposto

O processo de identificação proposto, consiste, em primeiro lugar, na classificação da impressão digital, a comparar, de modo a seleccionar, na base de dados, o conjunto de impressões, com as quais vai ser comparada. Após o processo de classificação (ver capítulo 4), a impressão digital, é sujeita aos processos de pré-processamento e extracção de características. Este processamento é idêntico ao aplicado às impressões digitais que constam na base de dados (processos descritos em detalhe no capítulo 5). No final destes processos, a impressão digital a comparar, encontra-se agora, na mesma representação que as impressões digitais que constam

## CAPÍTULO 6. IDENTIFICAÇÃO COM IMPRESSÕES DIGITAIS

na base de dados. É agora possível, comparar a impressão digital com a base de dados, utilizando o algoritmo proposto a seguir (no apêndice B, é fornecido um diagrama com todos os passos do algoritmo).

Seja  $I$  a impressão digital a comparar e  $BD$  a base de dados das impressões digitais. Após a impressão digital ser classificada, temos  $C$ , que corresponde à classe da mesma. Fazendo uma pesquisa em  $BD$ , por todas as impressões digitais da classe  $C$ , cria-se assim uma lista  $L$ , de todas as impressões digitais que é necessário comparar com  $I$ , de modo a tentar obter uma correspondência.

O fluxograma 6.1, ilustra os passos do algoritmo de comparação, proposto nesta dissertação. Os algoritmos 6.1 e 6.2, apresentam, em pseudo-código, o funcionamento dos blocos (1) e (2), marcados no fluxograma 6.1.

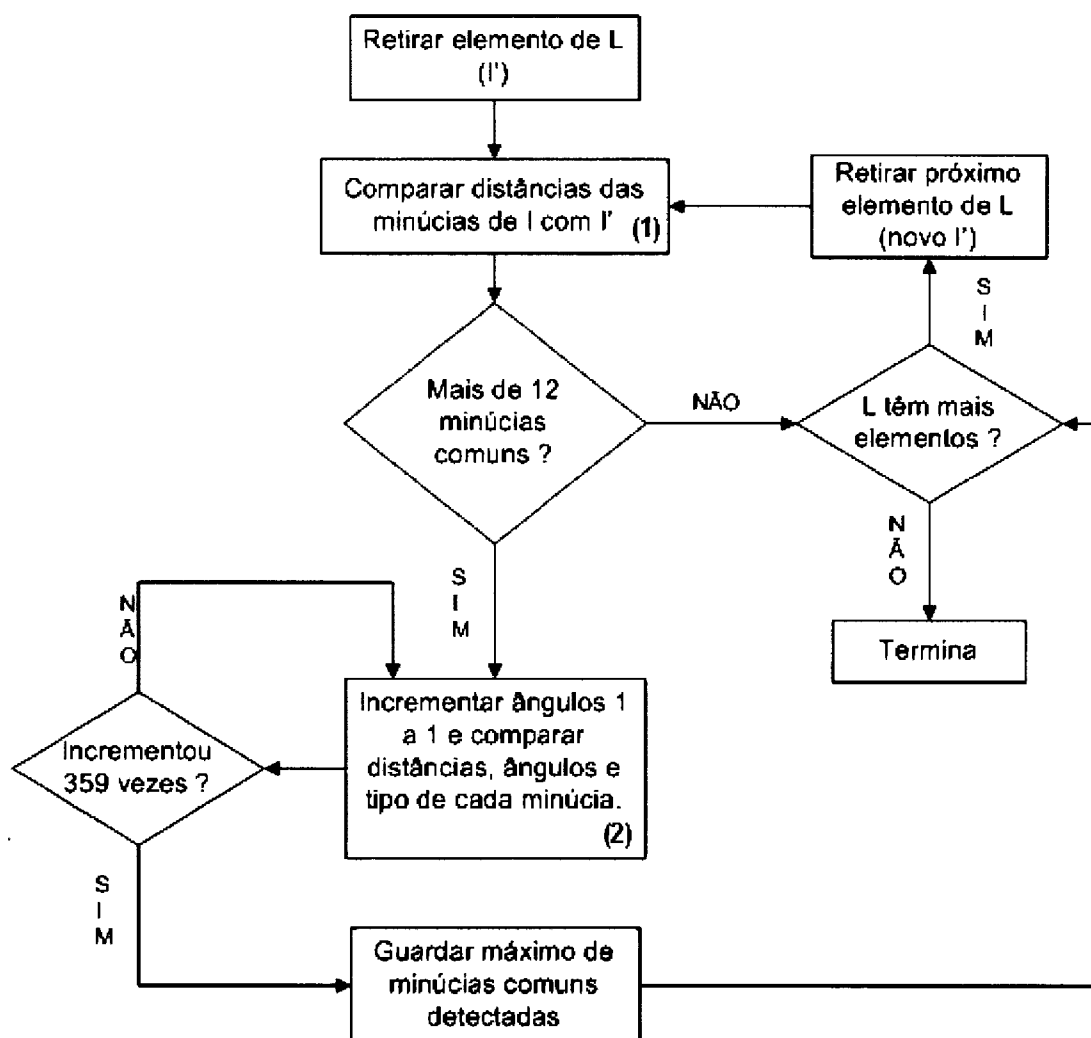


Figura 6.1 Fluxograma do algoritmo de comparação.

## 6.1. ALGORITMO DE COMPARAÇÃO PROPOSTO

Para cada impressão digital presente na lista calculada anteriormente ( $L$ ), é retirado uma imagem de cada vez ( $I'$ ). Para cada uma das imagens, são comparadas as distâncias das suas minúcias, com as distâncias das minúcias da impressão digital a comparar ( $I$ ). O algoritmo 6.1, representa este passo em pseudo-código, que corresponde ao bloco (1) do fluxograma 6.1. No referido algoritmo,  $idcmp$  e  $idbd$  correspondem, respectivamente, à representação da impressão digital a comparar ( $I$ ), e à representação da impressão digital de  $L$  actual. O valor do erro máximo considerado, é dado por  $err$ .

```
FUNCTION CalculaDistancias(idcmp, idbd, err)
  VARS STRUCTURE:
    idcmp, idbd <- [(Dst, Ang, Type), ...]
  START:
    mComuns <- 0
    for i <- 0 to idcmp.numberOfElements {
      for j <- 0 to idbd.numberOfElements {
        if (idbd[j].DST-err > idcmp[i].DST < idbd[j].DST+err)
          {
            mComuns++
            idcmp.removeElementAt(j)
          }
      }
    }
  END:
  return mComuns
END FUNCTION
```

**Algoritmo 6.1** Pseudo-código para o cálculo das minúcias comuns entre duas representações.

Caso o número de minúcias, cujas distâncias sejam idênticas (considerando uma margem de erro  $\epsilon$ ), seja menor que 12 (em 2.1.2, foi visto que são necessários pelo menos 12 pontos característicos comuns para que duas impressões digitais sejam consideradas iguais), então essa impressão digital é descartada, uma vez que não tem pontos suficientes em comum com a imagem a comparar ( $I$ ).

Caso o número de minúcias comuns entre as duas impressões digitais seja igual, ou superior a 12, é necessário comparar os seus ângulos e tipo de minúcia.

De forma a resolver o problema da rotação das impressões digitais, são efectuados diversos incrementos de  $1^\circ$ , de cada vez (até um máximo de  $359^\circ$ , que corresponde a uma rotação total

## CAPÍTULO 6. IDENTIFICAÇÃO COM IMPRESSÕES DIGITAIS

da impressão digital), a todos os ângulos das minúcias de I.

```
FUNCTION ComparaMinucias(idcmp, idbd, ed, ea)
  VARS STRUCTURE:
    idcmp, idbd ← [(Dst,Ang,Type),...]
  START:
    minuciasComuns, maxMinucias ← 0
    for i ← 0 to 359 {
      for k ← 0 to idcmp.numberOfElements {
        idcmp[k].Ang = idcmp[k].Ang+1
      }
      for l ← 0 to idcmp.numberOfElements {
        for j ← 0 to idbd.numberOfElements {
          if (idbd[j].DST-ed > idcmp[l].DST < idbd[j].DST+ed) {
            if (idbd[j].ANG-ea > idcmp[l].ANG < idbd[j].ANG+ea) {
              if (idbd[j].TYPE == idcmp[l].TYPE) {
                minuciasComuns++;
                idcmp.removeElementAt(j)
              }
            }
          }
        }
      }
      if (minuciasComuns > maxMinucias) {
        maxMinucias = minuciasComuns
      }
    }
  END:
  return maxMinucias
END FUNCTION
```

**Algoritmo 6.2** Pseudo-código para a comparação das minúcias.

Em cada um dos incrementos, são comparadas as distâncias, ângulos e tipo de minúcia, de todas as minúcias da impressão digital a comparar ( $I$ ), com a impressão digital em questão ( $I'$ ), guardando-se o valor máximo de minúcias comuns, entre as duas. Este processo, corresponde ao bloco (2) do fluxograma 6.1, e é apresentado em pseudo-código, no algoritmo 6.2. No

## 6.1. ALGORITMO DE COMPARAÇÃO PROPOSTO

---

referido algoritmo, *idcmp* e *idbd* correspondem, respectivamente, à representação da impressão digital a comparar (*I*), e à representação da impressão digital de *L*, actual. Em relação aos valores de erro máximos, tanto para a distância, como para o ângulo, são dados por *ed* e *ea*.

Terminada a rotação da impressão digital, é guardado, numa lista ordenada (*M*), o valor máximo de minúcias em comum, entre as duas impressões digitais, juntamente com a identificação da impressão digital em questão (*I'*). Proceda-se desta forma, para todas as impressões digitais que pertencem à lista *L*.

Terminadas as impressões digitais na lista *L*, a lista ordenada *M*, contém as impressões digitais em que foram detectados mais pontos característicos em comum com a impressão digital a comparar (*I*).

Resumindo, o funcionamento do algoritmo proposto, é dado pelos seguintes passos:

1. Classificar a impressão digital a comparar.
2. Submeter a impressão digital a comparar, aos processos de pré-processamento e extração de características.
3. Extrair para uma lista, todas as impressões digitais da base de dados, cuja classe corresponda à classe da impressão digital a comparar.
4. Para cada impressão digital, da lista anterior, comparar a distância das suas minúcias, com a distância das minúcias da impressão digital a comparar. Caso o número de minúcias, cujas distâncias iguais, seja inferior a 12, a impressão digital da lista é descartada.
5. Incrementar, 1º de cada vez, os ângulos das minúcias da impressão digital a comparar. A cada incremento, são comparados os ângulos, distâncias, e tipo de minúcia, com a impressão digital da lista. O valor máximo de minúcias comuns detectadas, entre as duas impressões digitais, é guardado numa lista.
6. Terminadas as impressões digitais da lista, a lista ordenada, contém as impressões digitais que apresentam maiores semelhanças, com a impressão digital a comparar.



## **6.2 Exemplo de aplicação do algoritmo proposto**

De seguida, apresenta-se um pequeno exemplo, da aplicação deste algoritmo. Para este exemplo, foram criadas 4 representações de impressões digitais, uma para comparação, e as restantes 3 para a criação de uma base de dados. As impressões digitais que pertencem à base de dados, apresentam as seguintes características:

- Existe uma impressão digital que têm menos de 13 minúcias em comum com as outras duas.
- Uma das impressões da base de dados, corresponde à impressão a comparar, com uma rotação de 5°.
- A impressão digital que resta, tem suficientes minúcias em comum para ser comparada, mas é diferente das outras duas.

A figura 6.2 representa uma possível impressão digital ( $F_C$ ) que se pretende comparar com uma base de dados, já classificada, processada e na representação proposta. A figura 6.3 representa uma base de dados, composta por 3 impressões digitais ( $FBD_1, FBD_2$  e  $FBD_3$ ), pertencentes à mesma classe que  $F_C$ .

```
[
[3.2,70,Terminação],    [18.1,41,Bifurcação],    [28.9,67,Terminação],    [36.2,2,Terminação],
[54.4,131,Terminação], [57.7,23,Bifurcação],    [60.8,88,Bifurcação],    [71.2,107,Bifurcação],
[83.4,140,Terminação], [90.9,31,Terminação],    [101.7,178,Bifurcação], [112.3,216,Terminação],
[118.5,307,Bifurcação], [123.1,25,Terminação],    [131.3,44,Bifurcação],    [149.5,73,Terminação],
[156.9,22,Bifurcação], [175.2,14,Terminação],    [182.6,129,Terminação], [198.3,19,Bifurcação],
[223.5,178,Terminação], [237.8,12,Bifurcação],    [243.3,169,Terminação], [286.6,186,Terminação],
[291.1,27,Terminação]
]
```

**Figura 6.2** Exemplo de uma representação de uma impressão digital.

Aplicando o algoritmo de comparação proposto, verifica-se que, após a contagem das minúcias que se encontram à mesma distância,  $FBD_1$  tem 0 minúcias à mesma distância,  $FBD_2$ ,

## 6.2. EXEMPLO DE APLICAÇÃO DO ALGORITMO PROPOSTO

15 minúcias à mesma distância, e finalmente,  $FBD_3$ , com 25 minúcias à mesma distância. Como  $FBD_1$  apresenta menos do que 13 minúcias à mesma distancia, esta impressão digital é descartada. Na figura 6.4, estão marcadas a cor verde, as minúcias que se encontram à mesma distância das minúcias de  $F_C$ .

Após excluídas as impressões digitais que não têm um mínimo de 13 minúcias com a mesma distância, é necessário comparar os ângulos e os tipos de minúcias.

<pre>[ [12.1,26,Terminação], [23.3,40,Bifurcação], [46.2,12,Terminação], [48.5,56,Bifurcação], [63.2,93,Terminação], [75.9,108,Terminação], [93.5,126,Terminação], [96.8,10,Bifurcação], [103.7,29,Bifurcação], [145.1,75,Terminação], [170.7,223,Terminação], [176.7,173,Bifurcação], [196.8,148,Bifurcação], [213.7,95,Terminação], ]</pre>	<pre>[ [5.4,,Terminação], [18.6,41,Bifurcação], [32.1,,Bifurcação], [54.8,13,Terminação], [67.2,,Terminação], [71.4,107,Bifurcação], [90.7,73,Terminação], [101.3,170,Terminação], [123.6,275,Bifurcação], [131.7,44,Terminação], [156.8,22,Bifurcação], [163.4,,Terminação], [182.2,129,Terminação], [198.7,189,Bifurcação], [223.6,178,Terminação], [237.2,121,Terminação], [243.8,169,Terminação], [260.2,,Terminação], [286.5,186,Terminação], [291.8,215,Terminação], ]</pre>	<pre>[ [3.2,12,Terminação], [28.7,72,Terminação], [54.8,18,Terminação], [60.2,93,Bifurcação], [83.9,145,Terminação], [101.3,183,Bifurcação], [118.6,312,Bifurcação], [131.7,49,Bifurcação], [156.6,27,Bifurcação], [182.2,134,Terminação], [223.6,183,Terminação], [243.8,174,Terminação], [291.8,32,Terminação], [18.6,46,Bifurcação], [36.4,7,Terminação], [57.8,28,Bifurcação], [71.4,112,Bifurcação], [90.7,78,Terminação], [112.4,221,Terminação], [123.6,280,Terminação], [149.1,78,Terminação], [175.4,19,Terminação], [198.7,174,Bifurcação], [237.2,126,Bifurcação], [286.5,191,Terminação], ]</pre>
(a) $FBD_1$	(b) $FBD_2$	(c) $FBD_3$

Figura 6.3 Base de dados com 3 impressões digitais pertencentes à mesma classe.

<pre>[ [5.4,,Terminação], [32.1,,Bifurcação], [67.2,78,Terminação], [90.7,73,Terminação], [123.6,275,Bifurcação], [156.8,22,Bifurcação], [182.2,129,Terminação], [223.6,178,Terminação], [243.8,169,Terminação], [286.5,186,Terminação], ]</pre>	<pre>[18.6,41,Bifurcação], [54.8,13,Terminação], [71.4,107,Bifurcação], [101.3,170,Terminação], [131.7,44,Terminação], [163.4,103,Terminação], [198.7,189,Bifurcação], [237.2,121,Terminação], [260.2,35,Terminação], [291.8,215,Terminação], ]</pre>	<pre>[3.2,12,Terminação], [28.7,72,Terminação], [54.8,18,Terminação], [60.2,93,Bifurcação], [83.9,145,Terminação], [101.3,183,Bifurcação], [118.6,312,Bifurcação], [131.7,49,Bifurcação], [156.6,27,Bifurcação], [182.2,134,Terminação], [223.6,183,Terminação], [243.8,174,Terminação], [291.8,32,Terminação], [18.6,46,Bifurcação], [36.4,7,Terminação], [57.8,28,Bifurcação], [71.4,112,Bifurcação], [90.7,78,Terminação], [112.4,221,Terminação], [123.6,280,Terminação], [149.1,78,Terminação], [175.4,19,Terminação], [198.7,174,Bifurcação], [237.2,126,Bifurcação], [286.5,191,Terminação], ]</pre>
(a) $FBD_2$	(b) $FBD_3$	

Figura 6.4 Minúcias com a mesma distância da impressão digital a comparar.

## CAPÍTULO 6. IDENTIFICAÇÃO COM IMPRESSÕES DIGITAIS

Para a impressão digital  $FBD_2$ , após uma rotação completa de  $F_C$ , e sucessivas comparações dos ângulos das minúcias, e posteriormente os tipos, conclui-se que o maior número de minúcias coincidentes, ocorreu sem qualquer incremento, ou seja, utilizando para comparação, os valores originais dos ângulos de  $F_C$ . Assim, após a comparação, verificou-se que,  $FBD_2$  apresenta 13 minúcias em comum com  $F_C$ . A figura 6.5 apresenta, pintadas a cor verde, as minúcias em comum com  $F_C$ , e a vermelho, as minúcias que se encontravam à mesma distância, mas cujo ângulo ou tipo não coincidiram com nenhuma minúcia de  $F_C$ .

[5.4,,Terminação],	[18.6,41,Bifurcação],	[32.1,,Bifurcação],	[54.8,13,Terminação],
[67.2,,Terminação],	[71.4,107,Bifurcação],	[90.7,73,Terminação],	[101.3,170,Terminação],
[123.6,275,Bifurcação],	[131.7,44,Terminação],	[156.8,22,Bifurcação],	[163.4,,Terminação],
[182.2,129,Terminação],	[198.7,189,Bifurcação],	[223.6,178,Terminação],	[237.2,121,Terminação],
[243.8,169,Terminação],	[260.2,,Terminação],	[286.5,186,Terminação],	[291.8,215,Terminação],

**Figura 6.5** Minúcias de  $FBD_2$  em comum com  $F_C$ .

Terminada a comparação com  $FBD_2$ , o algoritmo continua com a próxima impressão digital, que neste caso é  $FBD_3$ . Fazendo a rotação de  $F_C$ , verifica-se, após comparação dos ângulos e tipos, que o maior número de minúcias coincidentes, ocorreu quando se fez um incremento de  $5^\circ$  aos ângulos das minúcias de  $F_C$ . O número total de minúcias em comum entre as duas impressões digitais é de 19. Na figura 6.6, encontram-se pintadas a verde, as minúcias em comum com  $F_C$ , e a vermelho, as minúcias que se encontravam à mesma distância, mas cujos ângulos ou tipo não coincidiram com nenhuma outra.

[3.2,12,Terminação],	[18.6,46,Bifurcação],	[28.7,72,Terminação],	[36.4,7,Terminação],
[54.8,18,Terminação],	[57.8,28,Bifurcação],	[60.2,93,Bifurcação],	[71.4,112,Bifurcação],
[83.9,145,Terminação],	[90.7,78,Terminação],	[101.3,183,Bifurcação],	[112.4,221,Terminação],
[118.6,312,Bifurcação],	[123.6,280,Terminação],	[131.7,49,Bifurcação],	[149.1,78,Terminação],
[156.6,27,Bifurcação],	[175.4,19,Terminação],	[182.2,134,Terminação],	[198.7,174,Bifurcação],
[223.6,183,Terminação],	[237.2,126,Bifurcação],	[243.8,174,Terminação],	[286.5,191,Terminação],
[291.8,32,Terminação]			

**Figura 6.6** Minúcias de  $FBD_3$  em comum com  $F_C$ .

### 6.3. ORIGEM DO CÓDIGO UTILIZADO

---

Uma vez que não há mais impressões digitais na base de dados, o algoritmo finaliza, devolvendo uma lista de impressões digitais, ordenada de forma decrescente, pelo número de minúcias comuns.

$[FBD_3, FBD_2]$

**Figura 6.7** Lista resultante da aplicação do algoritmo de comparação de impressões digitais.

Para o exemplo em questão, o resultado do algoritmo seria equivalente ao da figura 6.7.

### 6.3 Origem do código utilizado

O código utilizado na implementação do algoritmo de comparação proposto, foi desenvolvido, na sua totalidade, recorrendo à linguagem de programação *JAVA*.

Na implementação deste algoritmo, não foi utilizado código de terceiros.

Para efectuar os cálculos matemáticos necessários, foi utilizado o pacote de matemática do *JAVA* (*java.Math*).



# Capítulo 7

## Aplicação Desenvolvida: Fingers

O objectivo proposto para este trabalho, foi efectuar um estudo sobre as técnicas necessárias para comparar impressões digitais entre si. De modo a fazer esse estudo, foi desenvolvida uma aplicação, constituída por um módulo responsável por introduzir as impressões digitais numa base de dados, e um outro que permite fazer a identificação de uma impressão digital nessa mesma base de dados. Os referidos módulos, implementam as técnicas apresentadas no capítulo 4 e 5, juntamente com o algoritmo para comparação das impressões digitais, proposto no capítulo 6. Na figura 7.1, apresenta-se um diagrama do funcionamento geral da aplicação desenvolvida.

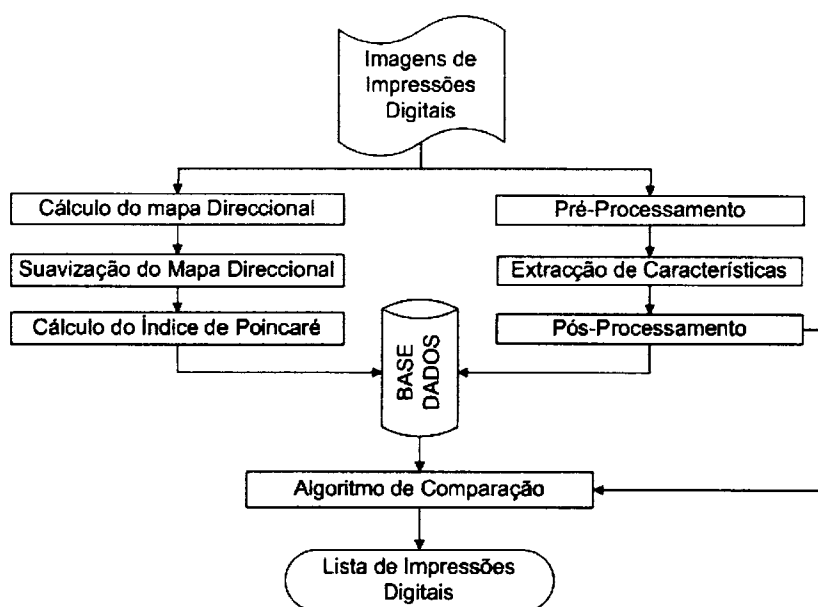


Figura 7.1 Diagrama da aplicação desenvolvida.

Como forma de testar o desempenho e utilidade das técnicas utilizadas, e do algoritmo de comparação proposto, a aplicação desenvolvida foi sujeita a diferentes testes, utilizando diferentes bases de dados de impressões digitais.

A aplicação foi desenvolvida e testada utilizando um computador portátil, cujas características podem ser consultadas na tabela 7.1.

<b>Características do computador</b>	
<b>Computador</b>	DELL Latitude 110L
<b>Processador</b>	Intel Celeron
<b>Velocidade Processador</b>	1.40 Ghz
<b>Memória RAM</b>	1.24 GB
<b>Sistema Operativo</b>	Alinex 3.9.10
<b>Java</b>	Versão 1.6.0
<b>GNU Octave</b>	Versão 3.2.2

**Tabela 7.1** Características do computador de desenvolvimento e testes.

Neste capítulo, são apresentadas as implementações utilizadas no desenvolvimento da aplicação, as diferentes bases de dados utilizadas para os testes, o conjunto de testes efectuados sobre a aplicação, assim como os respectivos resultados e discussões.

### 7.1 Descrição da base de dados

As imagens que formam a base de dados de impressões digitais, utilizadas nos testes da aplicação desenvolvida, foram seleccionadas das seguintes bases de dados, disponíveis para *download* na *internet*:

- FVC2000<sup>1</sup>, FVC2002<sup>2</sup> e FVC2004<sup>3</sup>

Este conjunto de impressões digitais é utilizado para os testes efectuados aos algoritmos propostos ao FVC. Estas bases de dados incluem impressões digitais capturadas através de sensores ópticos, e impressões digitais geradas por aplicações informáticas.

- CrossMatch<sup>4</sup>

---

<sup>1</sup>Download em <http://bias.csr.unibo.it/fvc2000/databases.asp>

<sup>2</sup>Download em <http://bias.csr.unibo.it/fvc2002/databases.asp>

<sup>3</sup>Download em <http://bias.csr.unibo.it/fvc2004/databases.asp>

<sup>4</sup>Download em <http://www.neurotechnology.com/fingerprint-scanner-cross-match-verifier-300-classic.html>

## 7.1. DESCRIÇÃO DA BASE DE DADOS

---

Esta base de dados é composta por um conjunto de impressões digitais, capturadas pelo leitor óptico *Cross Match Verifier 300*, com uma resolução de 500 dpi.

- UareU<sup>5</sup>

Este conjunto de impressões digitais foi capturado utilizando um leitor óptico *Digital-Persona U.are.U 4000*, com uma resolução de 500 dpi.

- Huelas<sup>6</sup>

Impressões digitais disponibilizadas sobre a forma de imagens, pela revista *online*, *Ciência Forense*<sup>7</sup>.

- SPD<sup>8</sup>

Conjunto de impressões digitais, utilizadas para testes no *Singular Point Detection Competition 2010*.

As impressões digitais que constituem estas bases de dados, foram obtidas sem qualquer preparação, instrução ou cuidado para garantir boa qualidade.

Base de dados	Nº de Impressões Seleccionadas
Huelas	41
SPD	65
CrossMatch	39
FVC2000	32
FVC2002	12
FVC2004	17
UareU	56
Total	262

**Tabela 7.2** Impressões digitais seleccionadas por base de dados.

De forma a testar a aplicação desenvolvida, foi construída uma base de dados de impressões digitais, baseada nas bases de dados acima descritas. Foi feita uma inspecção manual das impressões digitais, de modo a excluir impressões repetidas, sem pontos singulares visíveis, e

---

<sup>5</sup>Download em [http://www.neurotechnology.com/download/UareU\\_sample\\_DB.zip](http://www.neurotechnology.com/download/UareU_sample_DB.zip)

<sup>6</sup>Download em <http://www.esnips.com/web/DACTILOGRAMAS>

<sup>7</sup><http://www.cienciaforense.cl/>

<sup>8</sup>Download em [http://paginas.fe.up.pt/spd2010/test\\_dataset.zip](http://paginas.fe.up.pt/spd2010/test_dataset.zip)



## CAPÍTULO 7. APLICAÇÃO DESENVOLVIDA: FINGERS

algumas que eram apenas manchas negras. Na tabela 7.2, está referido o número de impressões digitais, seleccionadas de cada base de dados, num total de 262 impressões digitais diferentes.

Às impressões digitais seleccionadas para fazer parte da base de dados utilizada nos testes, foi previamente aplicado o módulo de classificação, como se pode observar na tabela 7.3. Para efeitos de testes, considerou-se esta primeira classificação como sendo correcta.

Classe	Huelas	SPD	Cmatch	FVC2000	FVC2002	FVC2004	UareU
Arco Plano	22	29	24	23	5	9	39
Verticilo	11	22	6	5	3	6	12
Arco Angular	0	1	0	1	0	0	1
Laço Esquerdo	4	10	4	1	2	1	3
Laço Direito	4	3	5	2	2	1	1
<b>Total</b>	41	65	39	32	12	17	56

**Tabela 7.3** Distribuição das impressões digitais seleccionadas, em classes.

## 7.2 Plataforma de desenvolvimento e implementações

A aplicação desenvolvida para testar os métodos e algoritmos propostos, foi implementada recorrendo à linguagem de programação *JAVA*<sup>9</sup>, devido à facilidade de construção de ambientes gráficos, integração com outras linguagens (uma vez que foram utilizados módulos desenvolvidos noutras linguagens de programação), manipulação de imagens e matrizes, e independência de plataforma.

O ambiente gráfico desenvolvido para a aplicação, foi implementado em *JAVA*, recorrendo aos pacotes *SWING* e *AWT*.

Como base de suporte para a aplicação, foi utilizada a aplicação de código aberto, *ImageJ*<sup>10</sup>, uma vez que a mesma, implementa os métodos necessários, para leitura e escrita de ficheiros de imagens, juntamente com alguns dos filtros necessários, para o sistema proposto. Os métodos, que constituem a aplicação *ImageJ*, são facilmente integrados em qualquer aplicação.

O processo de classificação das impressões digitais, é feito recorrendo a um módulo do projecto *Gaudi-AFIS*<sup>11</sup>, desenvolvido na linguagem de programação *C*, e integrado na aplicação.

<sup>9</sup>Copyright Oracle

<sup>10</sup><http://rsbweb.nih.gov/ij/download.html>

<sup>11</sup><http://sourceforge.net/projects/gaudi-afis/>

## 7.2. PLATAFORMA DE DESENVOLVIMENTO E IMPLEMENTAÇÕES

O projecto *Gaudi-AFIS*, foi desenvolvido, recorrendo ao trabalho de [22], cujos métodos são descritos no capítulo 4.3.

Os filtros de normalização de contraste, mediana, e esqueletização utilizados, estão implementados na aplicação *ImageJ*, referida anteriormente.

O algoritmo de cálculo do ponto de referência<sup>12</sup> utilizado, foi desenvolvido em *matlab*, por Luigi Rosa, tendo sido adaptado, de modo a funcionar com o *GNU Octave*. O filtro de gabor, pertence à biblioteca de funções *Fingerprint Verification System*<sup>13</sup>, desenvolvida na linguagem de programação *C*.

Os restantes filtros, e respectivo algoritmo de comparação, foram implementações próprias, desenvolvidas na linguagem de programação *JAVA*, com excepção do *Root Filtering*, que foi desenvolvido recorrendo ao *GNU Octave* (ver código fonte no apêndice D).

Todos os filtros desenvolvidos e utilizados respeitam as técnicas descritas em 5 e 6.

Origem do Código Fonte	Algoritmos Utilizados / Desenvolvidos
ImageJ	Métodos de leitura e escrita de imagens. Normalização de Contraste. Filtro de Mediana. Esqueletização.
Gaudi-AFIS	Módulo de classificação.
Luigi Rosa	Cálculo do ponto de referência.
FVS	Filtro de Gabor.
Desenvolvido para este trabalho	Ambiente gráfico. Ligação entre os diferentes algoritmos. Binarização adaptativa. Limpeza do esqueleto. Root Filtering. Crossing Number. Validação de minúcias. Construção da representação da impressão digital. Algoritmo de comparação de impressões digitais.

**Tabela 7.4** Origem do código fonte dos diversos algoritmos.

<sup>12</sup>[http://read.pudn.com/downloads166/sourcecode/graph/text\\_recognize/762700/centralizing.m\\_\\_.htm](http://read.pudn.com/downloads166/sourcecode/graph/text_recognize/762700/centralizing.m__.htm)

<sup>13</sup><http://fvs.sourceforge.net/>

Na tabela 7.4, é possível consultar, de forma estruturada, o origem dos diferentes algoritmos utilizados no desenvolvimento da aplicação proposta.

### 7.3 Arquitectura da aplicação

A aplicação desenvolvida, recorre a imagens de impressões digitais, previamente capturadas. Não existe nenhum tipo de limitação, no que se refere ao formato ou dimensões das imagens utilizadas. Prova disso, são as diferentes imagens, utilizadas nos testes, recolhidas de diversas bases de dados, disponíveis na *internet*.

A aplicação é composta por dois módulos: módulo de adição de impressões digitais à base de dados, e módulo de comparação de impressões digitais.

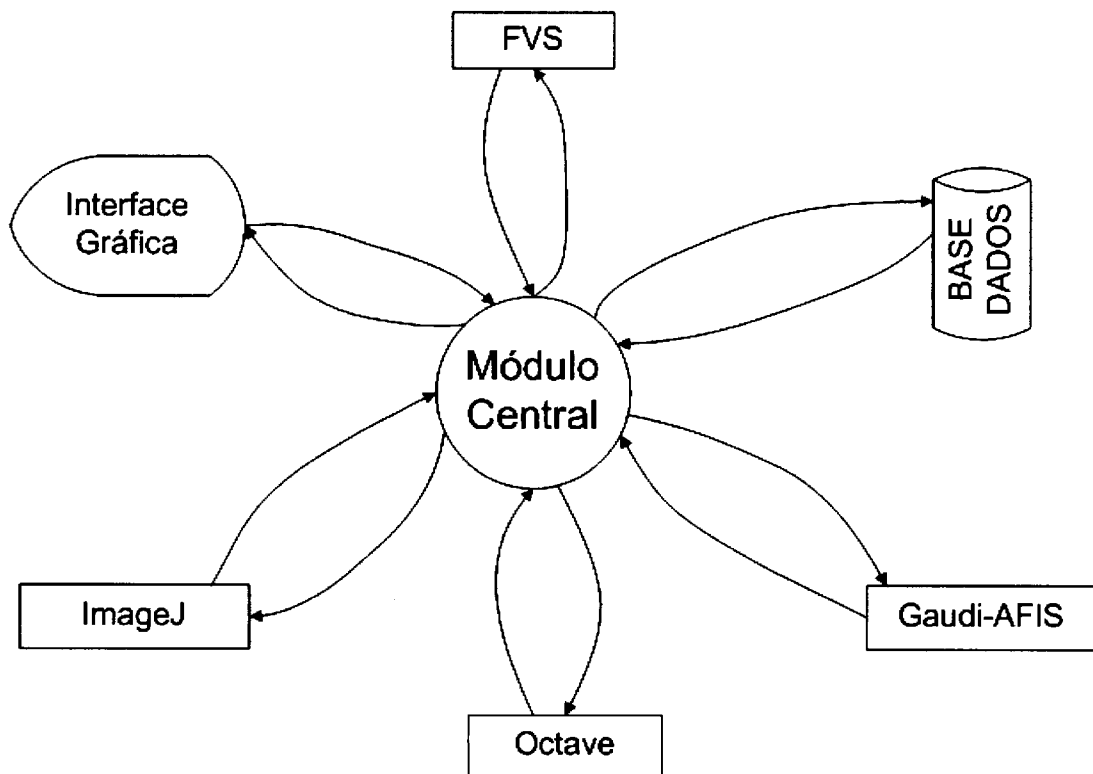


Figura 7.2 Arquitectura da aplicação desenvolvida.

A figura 7.2, representa a ligação, entre as diferentes aplicações, utilizadas no desenvolvimento da aplicação. A figura 7.3, mostra o fluxo de informação entre os diferentes módulos, que constituem a aplicação. Antes de ser possível comparar impressões digitais, é necessário, em primeiro lugar, adicionar impressões digitais à base de dados.

### 7.3. ARQUITECTURA DA APLICAÇÃO

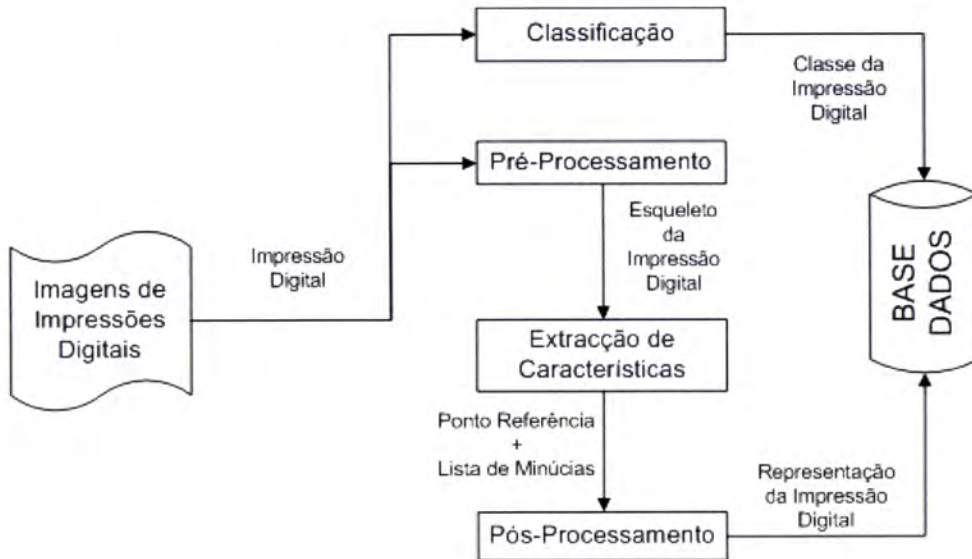


Figura 7.3 Fluxo de informação entre os módulos da aplicação.

#### 7.3.1 Ambiente Gráfico

A aplicação foi desenvolvida com um suporte gráfico, constituído por dois módulos principais: um módulo de adição de imagens à base de dados (ver figura 7.4), e um outro módulo (ver figura 7.5), que permite comparar uma impressão digital, com a base de dados previamente preenchida.

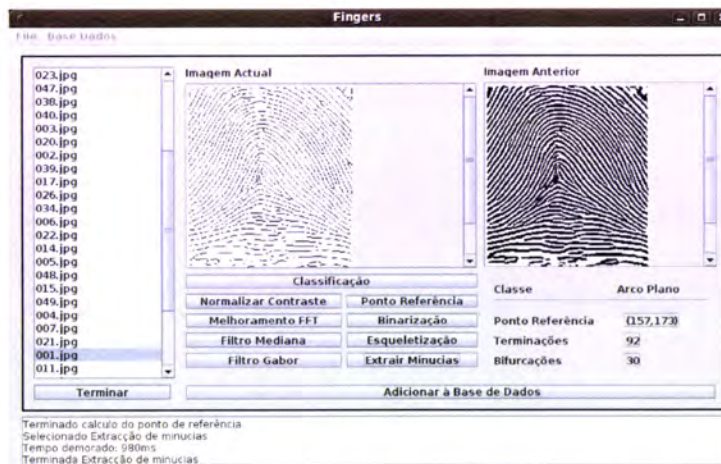


Figura 7.4 Módulo de adição de imagens à base de dados.

Cada um dos botões que constitui o ambiente gráfico, executa um algoritmo diferente.

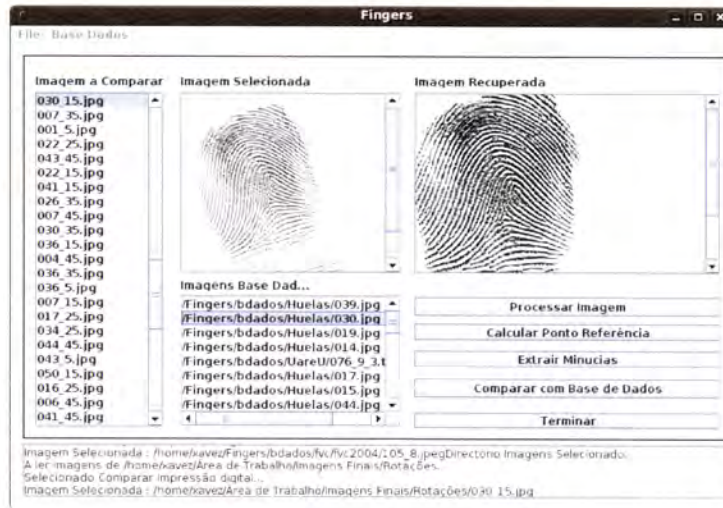


Figura 7.5 Módulo de identificação de impressões digitais.

### 7.3.2 Classificação

O primeiro passo, da aplicação, consiste em classificar a impressão digital a adicionar à base de dados. Esta classificação é feita, recorrendo ao módulo de classificação, da aplicação, Gaudi-AFIS.

A implementação desenvolvida consiste numa chamada ao sistema, com o nome da aplicação do Gaudi-AFIS, juntamente com o ficheiro da imagem da impressão digital. Após terminada a execução da mesma, é capturada a resposta da aplicação, ou seja, a classe a que corresponde a impressão digital.

### 7.3.3 Normalização do Contraste e Filtro de Mediana

Uma vez classificada a impressão digital, os próximos dois passos são a normalização do contraste, e a aplicação de um filtro de mediana. Para aplicar estes dois algoritmos, foi utilizada a implementação da aplicação ImageJ.

A normalização do contraste, é feita recorrendo ao método *equalize()*, acessível através do construtor *ij.plugin.ContrastEnhancer()*. À imagem resultante, da aplicação da normalização do contraste, é aplicado o filtro de mediana. Este filtro, é utilizado recorrendo ao construtor *ij.plugin.filter.RankFilters()*. Uma vez criado o construtor, é utilizado o método *filter.rank()*, cujos parâmetros são a imagem, o tamanho da janela a utilizar (foi utilizada uma janela de 4 pixeis) e o filtro em questão (filtro de mediana). No algoritmo 7.1, é visível o código necessário para utilizar estes dois filtros.

### 7.3. ARQUITECTURA DA APLICAÇÃO

---

```
// Abrir imagem da impressão digital
ImagePlus img = new ImagePlus(ImgImpressaoDigital);

// Normalização do contraste
ij.plugin.ContrastEnhancer cont
    = new ij.plugin.ContrastEnhancer();
cont.equalize(img);

// Filtro de mediana
ij.plugin.filter.RankFilters flt
    = new ij.plugin.filter.RankFilters();
filter.rank(img.getProcessor(), 2,
    ij.plugin.filter.RankFilters.MEDIAN);
```

**Algoritmo 7.1** Código necessário para utilizar os métodos do ImageJ.

#### 7.3.4 Root Filtering e Filtro de Gabor

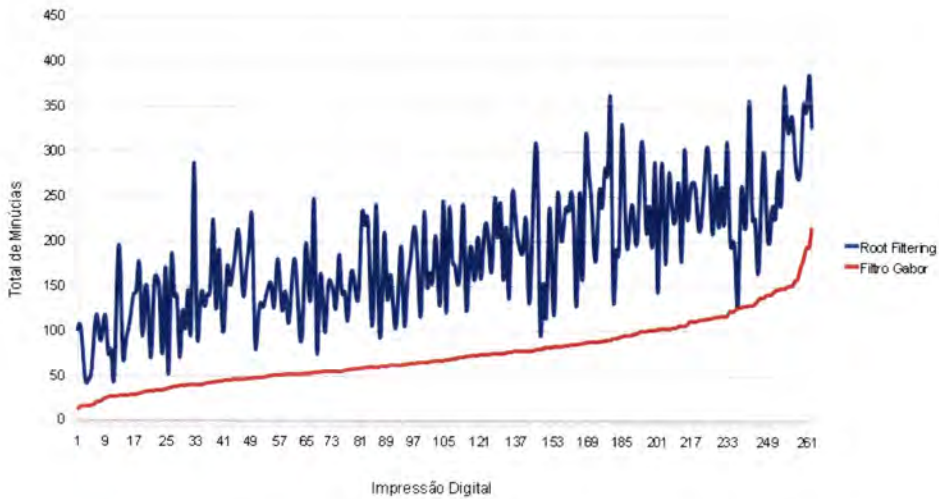
Foram implementados dois métodos para o melhoramento das cristas das impressões digitais: o *Root Filtering*, e o filtro de *Gabor*. Após a realização de vários testes, utilizando cada um deles independentemente, e de forma conjunta, verificaram-se melhores resultados, quando utilizado apenas o filtro de *Gabor*.

No gráfico 7.6, é possível verificar, através duma relação entre o total de minúcias extraídas, que o filtro de *Gabor* apresenta melhores resultados do que o *Root Filtering*. Apenas num caso pontual, o *Root Filtering* apresentou melhor resultado. Os valores do gráfico 7.6, encontram-se ordenados, pelo total de minúcias extraídas, utilizando o filtro de gabor.

Por este motivo, foi apenas utilizado o filtro de *Gabor* no melhoramento das cristas das impressões digitais, em todos os testes realizados.

A aplicação do filtro de *Gabor*, consiste em fazer uma chamada ao programa, passando como parâmetro, a imagem resultante, da aplicação do filtro de mediana. O resultado da aplicação do filtro de *Gabor*, é, uma nova imagem, que é necessário recarregar para a aplicação.

Apesar de não ter sido utilizado, a aplicação tem suporte para a utilização do root filtering. O filtro em questão, foi desenvolvido em Octave (no apêndice D, é possível consultar a implementação do filtro), e a sua utilização consiste, tal como o filtro de gabor, numa chamada ao sistema. A imagem é dividida em blocos de 64x64 pixels, sendo cada bloco, enviado individualmente ao Octave. O resultado é então capturado, e a imagem resultante reconstruída, bloco a bloco.



**Figura 7.6** Relação entre o Root Filtering e o Filtro de Gabor.

### 7.3.5 Binarização

Antes de aplicar o filtro de binarização à imagem otimizada pelo filtro de *Gabor*, é ajustado o seu tamanho (em altura e largura), de modo a que o mesmo seja múltiplo da janela utilizada, prevenindo assim, possíveis erros na aplicação. Após o acerto do tamanho da imagem, é aplicado o filtro, a blocos de 64x64 pixels. Terminada a binarização da imagem, é repostado o seu tamanho original. A implementação do filtro de binarização, passo-a-passo, pode ser vista a seguir.

1. Definir o tamanho ( $W$ ) do bloco (foram utilizados blocos de 64x64).
2. Ajustar a altura e largura da imagem, de modo a que as suas dimensões sejam múltiplas do bloco considerado.
3. Dividir a imagem em blocos de tamanho  $W$ .
4. Para cada bloco, calcular a média dos seus pixels.
5. Para cada pixel pertencente ao bloco em questão, caso o seu valor seja superior à média do bloco, substituir por 1, caso contrário, substituir por 0.
6. Repetir o passo anterior, até terminar o número de blocos da imagem.
7. Repôr o tamanho original da imagem.

## 7.3. ARQUITECTURA DA APLICAÇÃO

---

### 7.3.6 Esqueletização e respectiva limpeza

O próximo passo consiste em esqueletizar a imagem binarizada. O filtro de esqueletização utilizado encontra-se, também ele, disponível na aplicação ImageJ. Para utilizar o filtro em questão, é necessário, em primeiro lugar, recorrer ao construtor *ij.plugin.filter.Binary()*. Uma vez iniciado o construtor, recorre-se ao método *setup()*, informando que queremos utilizar a esqueletização na imagem fornecida, e, finalmente, aplica-se o método *run()*. No final deste processo, obtém-se uma imagem, cujo corpo contém apenas o seu esqueleto. No algoritmo 7.2, é possível consultar o código necessário para a utilização deste filtro.

```
ij.plugin.filter.Binary skel = new ij.plugin.filter.Binary();
skel.setup("skel", img);
skel.run(img.getProcessor());
```

**Algoritmo 7.2** Utilização do filtro de esqueletização do ImageJ.

Recorrendo à imagem esqueletizada, é agora feita, uma limpeza do esqueleto. A limpeza do esqueleto consiste em percorrer a imagem, em janelas de tamanho 3x3 pixeis, procurando pelos padrões anteriormente mencionados em 5.1.5.1.

### 7.3.7 Crossing Number e Validação de minúcias

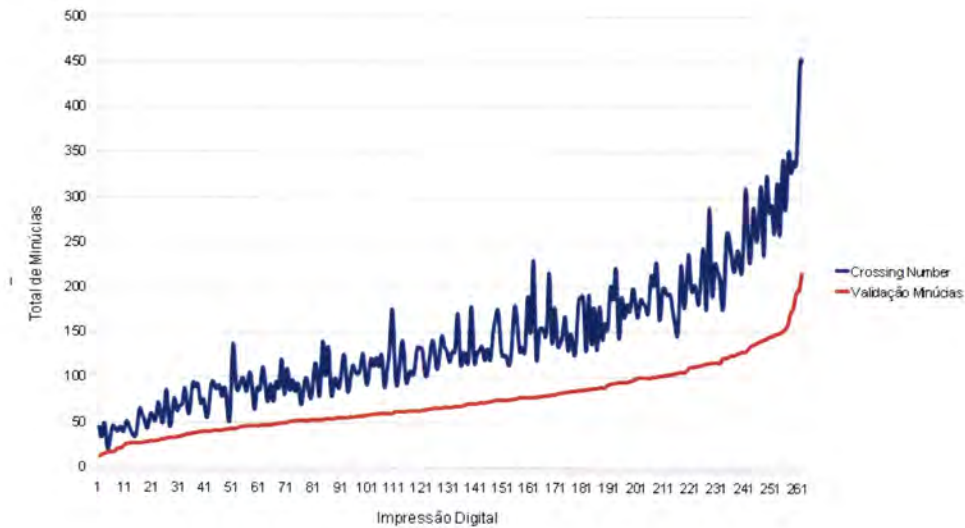
Finalizado o processamento da imagem, têm-se agora uma imagem de impressão digital, na forma de um esqueleto da mesma. Sobre este esqueleto, é agora aplicado o algoritmo de extracção de minúcias (CN).

O CN, consiste em percorrer a imagem, em janelas de tamanho 3x3, aplicando a fórmula do mesmo (ver capítulo 5.2.2), a cada uma das janelas. Sempre que é encontrado um ponto, que seja marcado como sendo uma minúcia, pelo algoritmo, são guardadas as suas coordenadas espaciais  $(x,y)$ , e o tipo de minúcia (terminação ou bifurcação) que foi detectada nessas coordenadas, numa lista.

Como o algoritmo CN, funciona com janelas bastante pequenas, são extraídas muitas falsas minúcias. De forma a minimizar o número de minúcias falsas, após a aplicação do algoritmo CN, é aplicada uma técnica de validação de minúcias à lista resultante. O gráfico 7.7 ilustra a relação entre o total de minúcias extraídas pelo algoritmo CN, e o número de minúcias que foram consideradas válidas, após a aplicação da técnica de validação. É possível verificar, que o número de minúcias consideradas válidas é agora bastante menor. Os valores do gráfico 7.7,



encontram-se ordenados pelo total de minúcias extraídas, utilizando o algoritmo de validação de minúcias.



**Figura 7.7** Relação entre o total de minúcias extraídas, com e sem validação.

Para aplicar a técnica de validação de minúcias, é utilizada uma janela de tamanho  $11 \times 11$ . Cada ponto  $(x, y)$ , de cada minúcia extraída pelo algoritmo CN, é centrado numa matriz  $11 \times 11$ . Nessa mesma matriz, são colocados todos os pontos que se encontram conectados ao ponto extraído. Desta forma, reproduzem-se apenas as linhas que originam a minúcia, tendo como origem o fim da mesma. A validação, consiste em verificar, se as linhas que dão origem à minúcia, têm comprimento suficiente para caber dentro da matriz. Caso isso não se verifique, as linhas são demasiado pequenas, provavelmente erros introduzidos pelos processos de pré-processamento.

### 7.3.8 Cálculo do ponto de referência

Para calcular o ponto de referência, é criado um ficheiro auxiliar do *GNU Octave*, que contém a chamada da função, que calcula o ponto de referência, juntamente com a imagem, para a qual vai ser calculado. Posteriormente à criação desse ficheiro temporário, é executado internamente pela aplicação, o *GNU Octave*, passando o ficheiro temporário como parâmetro. No final da execução do programa, é capturada a posição  $(x, y)$  do ponto de referência, detectada pela função.

## 7.3. ARQUITECTURA DA APLICAÇÃO

### 7.3.9 Construção da representação

Antes de se adicionar a impressão digital à base de dados, é necessário criar uma representação baseada na posição do ponto de referência. Assim, para cada minúcia detectada, é calculada a distância euclidiana até ao ponto de referência e o ângulo que faz com ele. O ponto de referência é considerado a origem de um sistema de eixos. À medida que vão sendo calculadas as distâncias e os ângulos, essa informação é guardada numa lista ordenada pela distância ao ponto de referência. A base de dados para as impressões digitais utilizada na aplicação, consiste numa tabela de *hash*, cujos elementos são 5 tabelas de *hash*, que representam as diferentes classes de impressões digitais.

Tabela de Hash Externa									
Tabela de Hash Arco Angular		Tabela de Hash Arco Plano		Tabela de Hash Laço Direito		Tabela de Hash Laço Esquerdo		Tabela de Hash Verticilo	
ID	REPRESENTAÇÃO	ID	REPRESENTAÇÃO	ID	REPRESENTAÇÃO	ID	REPRESENTAÇÃO	ID	REPRESENTAÇÃO
ID	REPRESENTAÇÃO	ID	REPRESENTAÇÃO	ID	REPRESENTAÇÃO	ID	REPRESENTAÇÃO	ID	REPRESENTAÇÃO
ID	REPRESENTAÇÃO	ID	REPRESENTAÇÃO	ID	REPRESENTAÇÃO	ID	REPRESENTAÇÃO	ID	REPRESENTAÇÃO
ID	REPRESENTAÇÃO	ID	REPRESENTAÇÃO	ID	REPRESENTAÇÃO	ID	REPRESENTAÇÃO	ID	REPRESENTAÇÃO
ID	REPRESENTAÇÃO	ID	REPRESENTAÇÃO	ID	REPRESENTAÇÃO	ID	REPRESENTAÇÃO	ID	REPRESENTAÇÃO
•	•	•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•	•	•

Figura 7.8 Representação da estrutura da base de dados.

A figura 7.8 representa a estrutura da base de dados utilizada. Na referida estrutura, cada *ID* é um identificador sequencial, e a *REPRESENTAÇÃO* corresponde à representação da impressão digital, da respectiva classe, que se pretende adicionar à base de dados.

### 7.3.10 Comparação de impressões digitais

Após preenchida a base de dados, é agora possível proceder a comparações entre impressões digitais. Para comparar uma impressão digital, com a base de dados, é necessário seleccionar a impressão digital a comparar. Após seleccionada, é aplicado a esta, todos os procedimentos anteriores, até ao ponto em que temos a representação da impressão digital. Recorrendo à classe da impressão digital a comparar, é extraída, da base de dados, a tabela de *hash* correspondente. Recorrendo à representação da impressão digital a comparar, e à tabela de *hash* que contém todas as impressões digitais, que pertencem à classe da mesma, é aplicado o algoritmo de comparação, descrito no capítulo 6. Os valores de erro máximo, considerados para minúcias coincidentes, tanto para as distâncias, como para os ângulos, após testes realizados, foram fixados em 5 pixels para a distância e 0.2° para o ângulo. Estes foram os valores que obtiveram

melhores resultados, uma vez que, valores mais baixos, resultavam na não identificação de nenhuma impressão digital (devido à necessidade de uma margem de erro, no cálculo do ponto de referência), e valores mais elevados, aumentava demasiado o número de impressões digitais erradamente consideradas semelhantes.

O resultado da comparação, é uma lista constituída pelas impressões digitais que mais se assemelham à impressão digital a comparar. No fim da comparação, essa lista é reduzida apenas às 20 impressões digitais mais semelhantes.

### 7.4 Complexidade temporal

De forma a registar o tempo que cada algoritmo necessita para terminar, foi introduzido, no código da aplicação, um contador. Na altura em que foi construída a base de dados, utilizada para as comparações, foi registado, individualmente, o tempo de cada algoritmo. A tabela 7.5 apresenta os tempos médios registados para cada algoritmo.

Algoritmo	Média(s)
Classificação	2,28
Normalização do contraste	0,03
Filtro de Mediana	0,09
Root Filtering	15,60
Filtro de Gabor	11,44
Cálculo do ponto de referência	66,14
Binarização	0,42
Esqueletização	1,67
Crossing Number	0,28
Validação das Minúcias	0,85
Identificação	5,07
<b>Total</b>	<b>103,87</b>

**Tabela 7.5** Tempo de execução de cada algoritmo.

Como se pode observar, pela análise da tabela anterior, todos os algoritmo são de execução bastante rápida, com excepção do algoritmo do cálculo do ponto de referência, devido à sua complexidade.

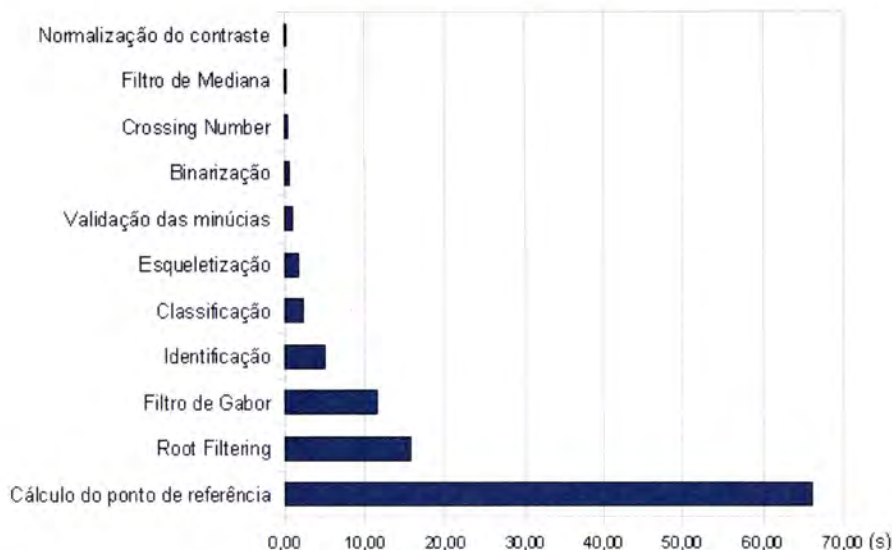
No gráfico 7.9, é possível visualizar, as diferenças de tempos de execução, entre os vários algoritmos.

O algoritmo de identificação demora aproximadamente 5 segundos a fazer uma comparação (independentemente da classe a que ela pertença), sendo este valor independente de todo

## 7.5. TESTES À APLICAÇÃO

---

o restante processamento. O tempo total necessário para identificar uma impressão digital, corresponde à soma total de todos os algoritmos, ou seja, aproximadamente 1,7 minutos.



**Figura 7.9** Tempo de execução dos algoritmos.

## 7.5 Testes à aplicação

De forma a verificar o desempenho do sistema proposto, foi feito um conjunto de testes à aplicação, utilizando diferentes imagens de impressões digitais. As imagens utilizadas para os testes à aplicação, foram feitas a partir das imagens originais, previamente adicionadas à base de dados.

Para os testes com as imagens originais, com ruído, deslocadas e parciais, foram criadas 50 imagens diferentes. No teste das imagens rotacionadas, foram utilizadas 100 imagens diferentes, com diversos ângulos de rotação.

Deste modo, foi criado o seguinte conjunto de imagens:

- Impressões digitais originais.  
Foi seleccionado um conjunto de imagens originais, de forma a verificar o correcto funcionamento do sistema proposto.
- Impressões digitais danificadas pelo ruído.  
De forma a testar os algoritmos de remoção de ruído e imperfeições, foi criado um conjunto de impressões digitais, aos quais foi acrescentado ruído, de forma aleatória.

- Deslocamento em relação à imagem original.  
Para verificar se o algoritmo de identificação proposto funciona para imagens deslocadas, foi criado um conjunto de imagens de impressões digitais, que se encontram deslocadas verticalmente e horizontalmente em relação às imagens originais.
- Rotação da impressão digital.  
Pretende-se que o algoritmo de identificação seja invariante à rotação, assim, criou-se um conjunto de imagens, nas quais as impressões digitais apresentam diferentes ângulos de rotação, em relação às imagens originais.
- Impressões digitais parciais.  
Para verificar o comportamento do algoritmo de identificação proposto, foi criado este conjunto de imagens de impressões digitais, às quais foram retirados pedaços da mesma, de forma aleatória. O único cuidado na remoção dos pedaços da impressão digital, foi a não remoção de zonas que continham o ponto de referência.

Para efeito de testes, considerou-se que uma impressão digital foi devidamente identificada com outra na base de dados, no caso de aparecer entre as 20 impressões digitais mais semelhantes. Os resultados dos testes efectuados, são apresentados a seguir, sob a forma de gráficos e tabelas. No apêndice E, encontram-se alguns exemplos das impressões digitais utilizadas nos testes à aplicação.

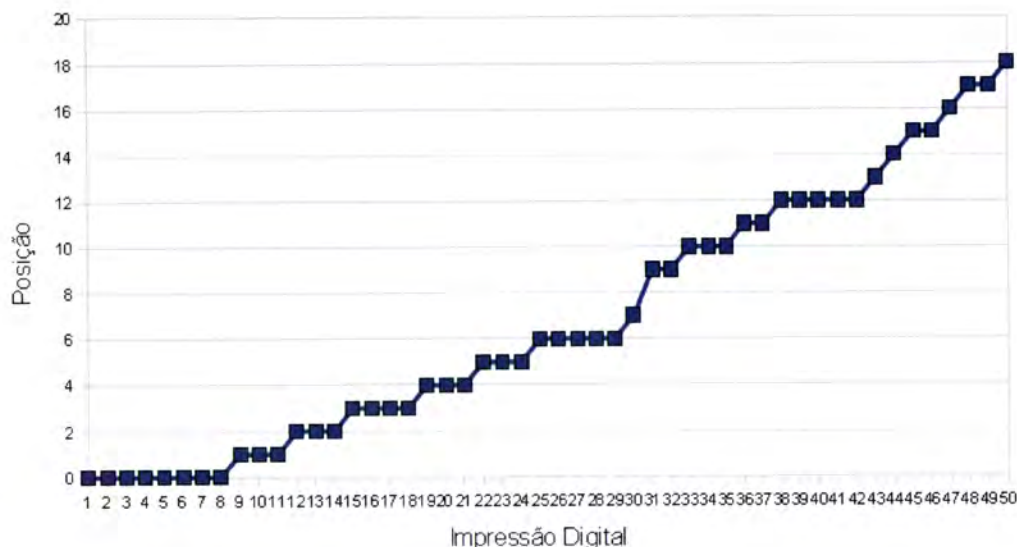
### 7.5.1 Impressões Digitais Originais

O primeiro teste efectuado à aplicação consiste em verificar o resultado da pesquisa das impressões digitais originais, na base de dados construída anteriormente. Foram testadas um total de 50 impressões digitais, seleccionadas aleatoriamente. A tabela 7.6 e o gráfico 7.10 mostram os resultados obtidos.

	Total	Percentagem
Não Identificado	8	16
Nos 5 primeiros	16	32
Entre 5ª e 10ª posição	11	22
Entre 10ª e 15ª posição	11	22
Entre 15ª e 20ª posição	4	8
<b>Total</b>	<b>50</b>	<b>100</b>

**Tabela 7.6** Resultados da identificação usando as imagens originais.

## 7.5. TESTES À APLICAÇÃO



**Figura 7.10** Distribuição da identificação das imagens originais.

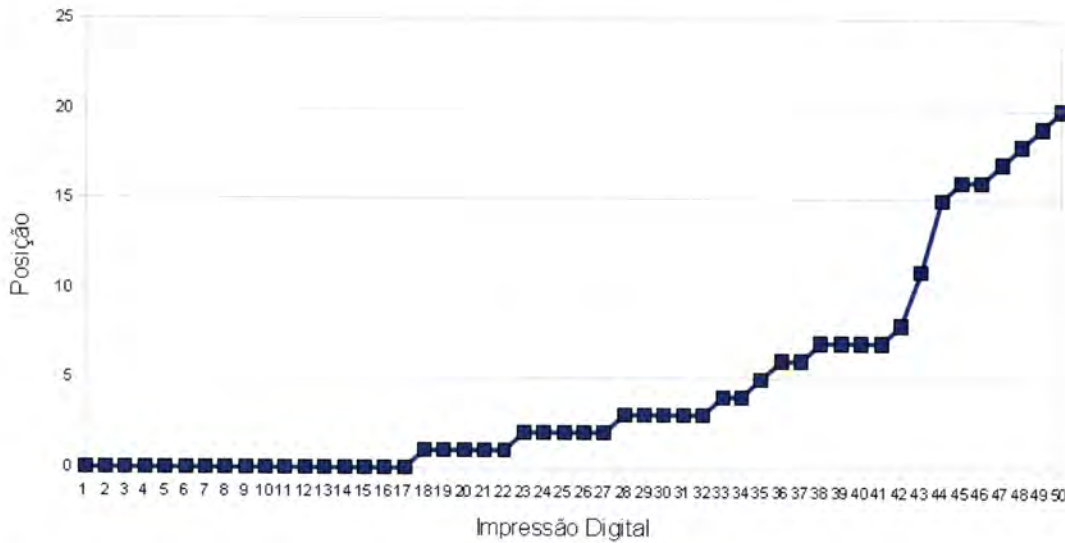
Como se pode observar, pelos resultados obtidos, apenas 8 das impressões digitais testadas não foram devidamente identificadas. Este resultado, deve-se ao facto, de ser necessário utilizar uma margem de erro no algoritmo de identificação. Essa margem de erro, faz com que impressões digitais que tenham maior número de minúcias, têm maior probabilidade de aparecer nos primeiros lugares do que as impressões que têm menor número de minúcias.

### 7.5.2 Impressões Digitais com Ruído

Para testar o funcionamento dos algoritmos de remoção de ruído e reconstrução de linhas, foi feito um teste à aplicação, utilizando imagens com às quais foi adicionado ruído, apesar das imagens originais já apresentarem bastantes deformações e ruído nas impressões digitais. Os resultados dos testes podem ser verificados na tabela 7.7 e no gráfico 7.11.

	Total	Percentagem
Não Identificado	17	34
Nos 5 primeiros	18	36
Entre 5ª e 10ª posição	7	14
Entre 10ª e 15ª posição	2	4
Entre 15ª e 20ª posição	6	12
<b>Total</b>	<b>50</b>	<b>100</b>

**Tabela 7.7** Resultados da identificação usando imagens danificadas com ruído.



**Figura 7.11** Distribuição dos resultados da identificação, usando imagens danificadas com ruído.

Os resultados do teste, mostram que 17 das impressões digitais testadas, não foram identificadas. Os erros de identificação detectados, devem-se, na sua maioria, a uma classificação errada, e a erro no cálculo do ponto de referência, como se pode observar na tabela 7.8.

	Total	Percentagem
Mal Classificadas	11	64,71
Erro Ponto Referência	6	35,29
<b>Total</b>	<b>17</b>	<b>100,00</b>

**Tabela 7.8** Erros verificados na identificação de imagens com ruído.

Apesar dos filtros de remoção de ruído e reconstrução de linhas funcionarem bem, na maioria dos casos, pode acontecer que o ruído existente danifique linhas essenciais para a correcta classificação, e localização do ponto de referência.

### 7.5.3 Impressões Digitais com Deslocamento

O teste com imagens de impressões digitais, que apresentam deslocamento em relação às imagens originais, têm como objectivo verificar o correcto funcionamento do algoritmo de identificação proposto, uma vez que o objectivo do mesmo era ser invariante ao deslocamento e à rotação. Assim, criou-se um conjunto de imagens, cujas impressões digitais foram deslocadas, de forma aleatória, horizontalmente e verticalmente.

## 7.5. TESTES À APLICAÇÃO

	Total	Percentagem
Não Identificado	20	40
Nos 5 primeiros	9	18
Entre 5ª e 10ª posição	10	20
Entre 10ª e 15ª posição	8	16
Entre 15ª e 20ª posição	3	6
<b>Total</b>	<b>50</b>	<b>100</b>

**Tabela 7.9** Resultados da identificação de impressões deslocadas.

A tabela 7.9 e o gráfico 7.12 mostram os resultados do teste realizado.



**Figura 7.12** Distribuição dos resultados do teste com imagens deslocadas.

Os resultados obtidos mostram que 20 das impressões digitais submetidas a teste, não foram correctamente identificadas com nenhuma outra da base de dados. Este resultado deve-se na sua maioria a erros de classificação, uma vez que 9 das impressões digitais não identificadas, foram mal classificadas.

	Total	Percentagem
Mal Classificadas	9	45
Ponto Referencia Errado	5	25
Outro Motivo	6	30
<b>Total</b>	<b>20</b>	<b>100</b>

**Tabela 7.10** Erros verificados na identificação de impressões deslocadas.

Das restantes impressões não identificadas, 5 resultaram de erro no cálculo do ponto de referência, e 6 devido às margens de erro do algoritmo de comparação. Na tabela 7.10, é



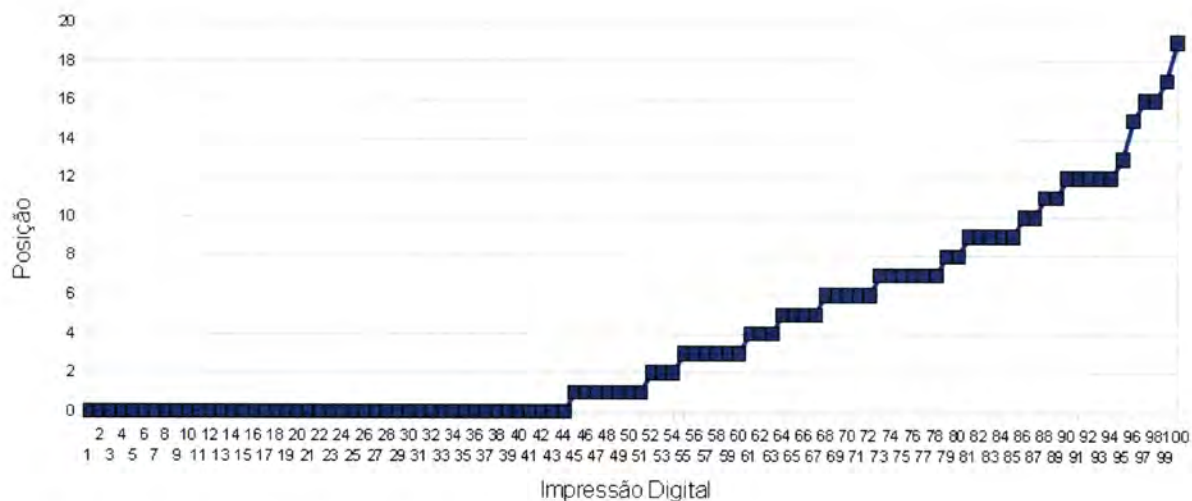
possível ver os erros das impressões digitais não identificadas.

### 7.5.4 Impressões Digitais com Rotações

Este conjunto de imagens de impressões digitais, tem como objectivo, assim como o anterior, testar o funcionamento da invariância à rotação, do algoritmo de identificação. Assim, foram criadas imagens de impressões digitais, com rotações de diferentes ângulos. A tabela 7.11 e o gráfico 7.13 mostram os resultados do teste efectuado.

	Total	Percentagem
Não Identificado	44	44
Nos 5 primeiros	23	23
Entre 5ª e 10ª posição	20	20
Entre 10ª e 15ª posição	9	9
Entre 15ª e 20ª posição	4	4
<b>Total</b>	<b>100</b>	<b>100</b>

**Tabela 7.11** Resultados da identificação de impressões digitais com rotações.



**Figura 7.13** Distribuição dos resultados da identificação com impressões que apresentam rotações.

Neste teste, verificou-se o maior número de impressões digitais não identificadas. No entanto, como se pode verificar na tabela 7.12, 43 das 44 impressões digitais não identificadas, resultaram de uma classificação errada. Apenas uma das impressões digitais não identificada resultou da margem de erro do algoritmo de identificação.

## 7.5. TESTES À APLICAÇÃO

	Total	Percentagem
Mal Classificadas	43,00	97,73
Outro Motivo	1,00	2,27
<b>Total</b>	<b>44,00</b>	<b>100</b>

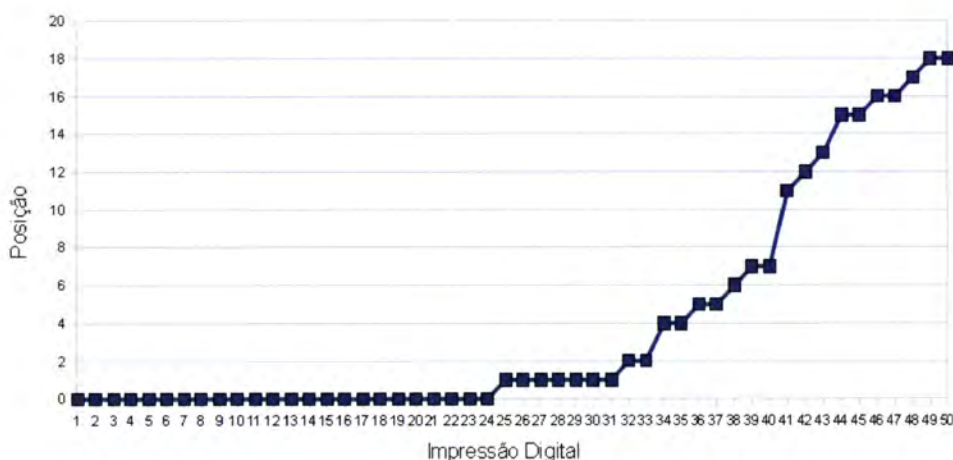
**Tabela 7.12** Erros verificados na identificação com impressões rotacionadas.

### 7.5.5 Impressões Digitais Parciais

O último teste efectuado à aplicação consiste em imagens de impressões digitais parciais. Este teste pretende verificar o comportamento da aplicação, quando se compara uma impressão digital, à qual foi retirada grande parte dos seus pontos característicos. Para tal, foi construído um conjunto de impressões digitais, às quais foram retirados pedaços, tentando apenas não retirar zonas que pudessem ser identificadas como a zona do ponto de referência. A tabela 7.13 e o gráfico 7.14 mostram os resultados do teste.

	Total	Percentagem
Não Identificado	24	48
Nos 5 primeiros	13	26
Entre 5ª e 10ª posição	3	6
Entre 10ª e 15ª posição	5	10
Entre 15ª e 20ª posição	5	10
<b>Total</b>	<b>50</b>	<b>100</b>

**Tabela 7.13** Resultados da identificação usando imagens de impressões parciais.



**Figura 7.14** Distribuição dos resultados do teste com impressões digitais parciais.

## CAPÍTULO 7. APLICAÇÃO DESENVOLVIDA: FINGERS

Como se pode observar nos resultados apresentados, 24 das impressões digitais criadas para este teste não foram identificadas. Mais uma vez, como se pode comprovar na tabela 7.14, a classificação errada das impressões digitais foi a maior causa desta não identificação, com um total de 15 impressões digitais mal classificadas.

	Total	Percentagem
Mal Classificadas	15	62,5
Ponto Referência Errado	1	4,20
Falta Minúcias	8	33,33
<b>Total</b>	<b>24</b>	<b>100</b>

**Tabela 7.14** Erros verificados na identificação de impressões parciais.

Os restantes factores foram, a falta de minúcias para comparação, num total de 8 impressões digitais com este problema, e 1 impressão digital, na qual foi incorrectamente detectado o ponto de referência.

### 7.5.6 Imagens Mal Classificadas

Como se pode verificar, nos testes anteriores, o elevado número de impressões digitais, não identificadas, deveu-se a erros de classificação. De modo a comprovar a influência da classificação errada, efectuou-se um teste recorrendo apenas a essas imagens.

Neste teste, foram seleccionadas todas as imagens de impressões digitais, usadas anteriormente, que resultaram numa não identificação.

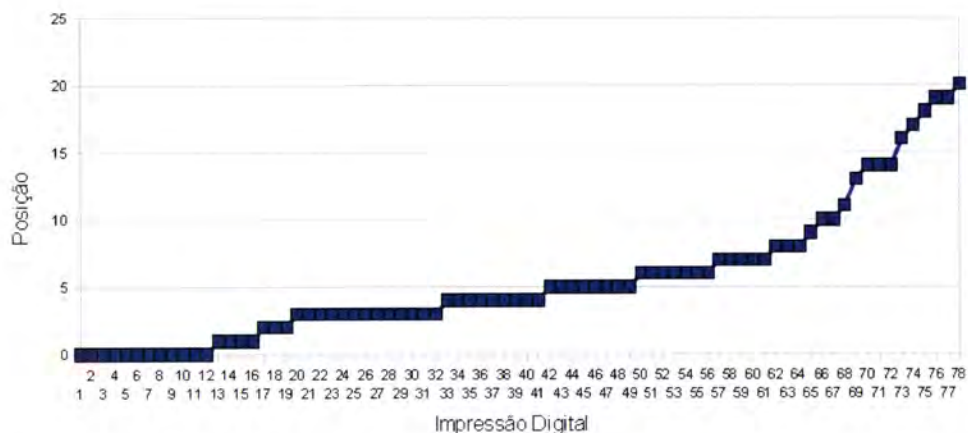
De modo a evitar o processo de classificação, este foi desactivado na aplicação. As imagens das impressões digitais foram divididas em grupos, consoante a sua classe. Cada grupo, foi então comparado com as impressões digitais, de classe equivalente, presentes na base de dados.

A tabela 7.15, e o gráfico 7.15, ilustram os resultados obtidos para este teste.

	Total	Percentagem
Não Identificado	12	15
Entre o 1 e o 5	37	47
Entre o 5 e o 10	18	23
Entre o 10 e o 15	5	6
Entre o 15 e o 20	6	8
<b>Total</b>	<b>78</b>	<b>100</b>

**Tabela 7.15** Resultados da identificação usando as imagens mal classificadas.

## 7.6. CONSIDERAÇÕES FINAIS



**Figura 7.15** Distribuição dos resultados da identificação das imagens mal classificadas.

Como se pode observar pelos resultados obtidos, apenas 12 das 78 impressões digitais submetidas a teste, não foram identificadas. O único motivo da falha na identificação destas 12 impressões digitais, foi a incorrecta localização do ponto de referência.

Com base nestes resultados, conclui-se que o processo de classificação utilizado, é o maior responsável pela não identificação das impressões digitais.

## 7.6 Considerações Finais

Os testes realizados sobre a aplicação desenvolvida, mostram que foi possível identificar 197 das 300 impressões digitais testadas. Este valor, corresponde a uma taxa de acerto de aproximadamente 66%. Dos algoritmos implementados no sistema proposto, conclui-se que, apenas o módulo de classificação apresenta um desempenho muito baixo, uma vez que, das 113 impressões digitais não identificadas, 78 não o foram, devido a erros de classificação, o que se traduz numa percentagem de aproximadamente 70%. Através do teste efectuado em 7.5.6, é possível afirmar que o motivo da não identificação destas impressões digitais, foi a classificação errada. As tabelas 7.16 e 7.17, mostram, respectivamente, o total de impressões identificadas e não identificadas, e os respectivos erros ocorridos.

Nº Impressões Digitais	
Identificadas	197
Não Identificadas	113
Total	300

**Tabela 7.16** Total de identificações correctas e incorrectas.

## CAPÍTULO 7. APLICAÇÃO DESENVOLVIDA: FINGERS

As restantes falhas de identificação, deveram-se a falhas no cálculo do ponto de referência (aproximadamente 11%), e à margem de erro utilizada no algoritmo de identificação (aproximadamente 13%).

	Nº Impressões Digitais
Erro Classificação	78
Ponto de Referência	12
Outro	15
Total	113

**Tabela 7.17** Total de erros verificados na identificação.

O módulo de classificação utilizado, foi desenvolvido com base no trabalho de [22]. Nesse mesmo trabalho, sugere-se a utilização de um algoritmo de segmentação, de modo a separar a impressão digital do fundo da imagem, antes de se proceder aos métodos necessários para a sua classificação. Este processo de segmentação, não se encontra a funcionar no módulo utilizado, o que poderá explicar os baixos valores de acerto do mesmo.

Nas imagens criadas para as impressões digitais deslocadas, de modo a criar o efeito de deslocamento, foi necessário acrescentar espaço em branco à imagem. Este acrescento de espaço, aliado ao facto do módulo não implementar o algoritmo de segmentação sugerido, pode, por um lado, induzir o método de suavização em erro, e, assim, originar a criação de zonas que sejam incorrectamente detectadas como pontos singulares pelo índice de poincaré.

No caso das imagens com rotações, o erro do método de classificação foi também bastante elevado. Apesar de nestas imagens, também ter sido necessário acrescentar espaço em branco, a rotação da imagem da impressão digital, altera os ângulos dos pontos singulares, induzindo assim, o cálculo do índice de poincaré em erro, e consequentemente a uma classificação errada.

# Capítulo 8

## Conclusões e trabalho futuro

O objectivo proposto para este trabalho, foi efectuar um estudo sobre as técnicas e métodos necessários para comparar impressões digitais, estudo esse feito sob a forma de desenvolvimento de uma aplicação informática.

A escolha dos algoritmos e métodos, assim como os seus parâmetros, utilizados na implementação do sistema proposto, foi feita recorrendo a diversa bibliografia sobre os problemas de classificação, melhoramento e comparação de impressões digitais.

O sistema de identificação de impressões digitais proposto assenta sobre um conjunto de técnicas necessárias para o aperfeiçoamento da imagem, conjugadas com algoritmos de extracção de características. Os pontos característicos, juntamente com um ponto de referência, foram utilizados para criar uma representação da impressão digital, que, recorrendo ao algoritmo de comparação desenvolvido, se torna invariante à rotação e ao deslocamento.

Os resultados obtidos nos testes realizados (ver capítulo 7.5), apesar de apresentarem um número elevado de impressões digitais que não foi possível identificar (apenas pela baixa eficácia demonstrada pelas técnicas de classificação), são bastante optimistas, uma vez que demonstram o correcto funcionamento dos métodos propostos, juntamente com o algoritmo de identificação desenvolvido. Os dois maiores problemas relativos à comparação de impressões digitais (deslocamento e rotações das mesmas) podem ser resolvidos recorrendo aos métodos propostos.

Com este estudo, foi possível concluir que o processo de comparação de impressões digitais, é altamente complexo, e dependente da qualidade das imagens. Nas imagens de impressões digitais em que a qualidade da imagem é relativamente boa, a facilidade de localizar os pontos característicos é maior, originando uma extracção dos mesmos mais fiável, e como consequência, uma identificação mais eficaz. A grande dependência de imagens de impressões digitais de boa qualidade, para o correcto funcionamento do sistema, obriga à utilização de

## CAPÍTULO 8. CONCLUSÕES E TRABALHO FUTURO

---

algoritmo de melhoramento de imagens, que consomem tempo, e nalguns casos, não produzem as melhorias necessárias para obter impressões digitais óptimas.

Como continuidade deste trabalho, tendo em vista o melhoramento do desempenho do sistema, sugerem-se as seguintes modificações:

- Modificação da implementação do algoritmo de cálculo do ponto de referência, de modo a tentar diminuir o tempo de processamento do mesmo.
- Implementação de todos os algoritmos na mesma linguagem de programação, com o objectivo de eliminar as chamadas ao sistema, diminuindo assim o tempo de processamento da aplicação.
- Alteração do mecanismo de classificação utilizado, com o objectivo de aumentar a taxa de sucesso da mesma.

Em relação às técnicas de classificação propostas, verifica-se que, apesar de apresentarem bons resultados para imagens de impressões digitais, que tenham sido capturadas direitas, o mesmo não acontece quando as impressões digitais sofrem algum tipo de rotação. A alteração do sistema de classificação proposto neste trabalho, por um outro que fosse invariante à rotação, iria aumentar bastante a taxa de sucesso de identificação das impressões digitais.

Em [58], é proposta a utilização de filtros complexos para a localização dos pontos singulares, em imagens de impressões digitais, de modo a poderem ser utilizados para a sua classificação, no entanto, não foram encontradas referências a nenhum trabalho que utilizasse estas técnicas, para classificar as imagens de impressões digitais.

# Glossário

<b>Antropometria</b>	Conjunto de técnicas utilizadas para medir o corpo humano ou as suas partes.
<b>Binarização</b>	Conversão de uma imagem em níveis de cinza para uma imagem com representação binária, ou seja, uma imagem com apenas brancos e pretos.
<b>Binarização Adaptativa</b>	Método de binarização no qual a imagem é dividida em blocos, e o limiar de binarização é calculado através de uma medida estatística para cada bloco.
<b>Biometria</b>	Medida de características físicas ou comportamentais das pessoas como forma de identificá-las unicamente.
<b>Classificação</b>	Distribuição por classes.
<b>Contraste</b>	O contraste define-se como sendo a diferença relativa da intensidade de um pixel de uma imagem com os seus vizinhos.
<b>Convolução</b>	A convolução é um operador que, a partir de duas funções, produz uma terceira.
<b>Cristas</b>	Numa imagem de impressão digital, as cristas representam as linhas pretas correspondentes às linhas dactilares.
<b>Cristas papilares</b>	Relevos epidérmicos situados na palma das mãos e na planta dos pés.



---

<b>Dactiloscopia</b>	Processo de identificação humana por meio das impressões digitais.
<b>Deltas</b>	Ângulo ou triângulo formado pelas cristas papilares. Pode ser formado de duas maneiras: pela bifurcação de uma linha simples ou pela divergência brusca de duas linhas paralelas.
<b>Distância Euclidiana</b>	A distância euclidiana corresponde à distância entre dois pontos $(x,y)$ .
<b>Esqueletização</b>	Processo de transformação de uma imagem binária em outra que representa o seu esqueleto, sem perder a forma estrutural da imagem original.
<b>Histograma</b>	É uma representação gráfica da distribuição de frequências de um conjunto de medições. No caso de imagens em tons de cinza, representa a distribuição dos valores das intensidades de todos os pixels, pelos valores da escala de cinzentos.
<b>Histograma Uniforme</b>	Histograma no qual todos os níveis apresentam a mesma frequência.
<b>Imagem Binária</b>	Imagens binárias são imagens cujos pixels apenas apresentam dois valores possíveis de intensidade, resultando assim em imagens com as cores preto e branco. Em termos numéricos, os valores de cada pixel são 0 para a cor preto e 1 ou 255 para a cor branco.
<b>Impressões Digitais</b>	Desenho formado pelas papilas digitais, presentes nas pontas dos dedos e das mãos, deixado em uma superfície lisa.
<b>Limiar de Binarização</b>	Valor através do qual se decide se um dado pixel, se transforma na cor preta ou branca.
<b>Lofoscopia</b>	Ciência que estuda os desenhos formados pelas cristas papilares.

<b>Mapa Direccional</b>	Mapa resultante do cálculo da direcção das linhas dactilares.
<b>Mediana</b>	A mediana é uma medida estatística que caracteriza as observações de uma determinada variável ordenada, de tal forma que a mesma separe a metade superior, da metade inferior da variável.
<b>Minúcias</b>	Pontos característicos das impressões digitais, compostos por bifurcações e terminações.
<b>Moda</b>	A moda é uma medida estatística, que representa o valor que detém o maior número de observações, ou seja, o valor com maior frequência.
<b>Núcleos</b>	Região central de uma impressão digital.
<b>Pelmatoscopia</b>	Ciência que estuda os desenhos formados pelas cristas papilares das palmas dos pés.
<b>Pixel</b>	Um pixel é o menor ponto que forma uma imagem digital, sendo que o conjunto de milhares de pixels formam a imagem inteira.
<b>Ponto de Referência</b>	Consiste num ponto de uma impressão digital, que se consiga distinguir de todos os outros, e calcular de forma não ambígua.
<b>Quiroscopia</b>	Ciência que estuda os desenhos formados pelas cristas papilares das palmas das mãos.
<b>Sistema de Bertillon</b>	Sistema antropométrico desenvolvido por Alphonse Bertillon. Consistia num sistema de identificação de criminosos, através da anotação das suas características e das medições de partes do corpo.
<b>Suavização</b>	Nome do processo que visa normalizar as direcções de um mapa direccional.

<b>Tabela de Hash</b>	É uma estrutura de dados, que associa chaves de pesquisa a valores. O seu objectivo é, a partir de uma chave simples, fazer uma busca rápida e obter o valor associado desejado.
<b>Tangente</b>	A tangente define-se como sendo a proporção entre o cateto oposto a um ângulo $\theta$ , e o cateto adjacente a esse mesmo ângulo.
<b>Transformada de Fourier</b>	É uma transformada integral que expressa uma função em termos de funções sinusoidais multiplicadas por coeficientes.
<b>Transformada Rápida de Fourier</b>	Versão da transformada de Fourier que utiliza valores discretos, para uso computacional ou de processamento digital de sinais.
<b>Vales</b>	Numa imagem de impressão digital, os vales correspondem aos intervalos a branco entre as linhas dactilares.
<b>Valores Discretos</b>	Chamam-se valores discretos, valores que representam uma quantidade finita.
<b>Variância</b>	É uma medida que indica quão longe os valores de uma dada variável se encontram do valor esperado.

# Referências Bibliográficas

- [1] A. Simas and F. Calisto. *Metodologias de investigação criminal: Lofoscopia*. Centro de Recursos Didáticos e Audiovisuais, 1999.
- [2] Sub núcleo de investigação criminal. Curso de investigação criminal para investigadores: A lofoscopia. 2000.
- [3] T. P. F. Costa. A importância do médico dentista na identificação de cadáveres em desastres de massa. Master's thesis, Faculdade de Ciências da Saúde - Universidade Fernando Pessoa, 2009.
- [4] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar. *Handbook of Fingerprint Recognition*. Springer-Verlag, 2º edition, 2009.
- [5] E. R. Henry. *Classification and uses of fingerprints*. George Routledge and Sons, 1900.
- [6] M. J. L. Piñol. Análisis del algoritmo de maio para la extracción de huellas dactilares. optimización para una implementación hardware. Master's thesis, Universitat Rovira I Virgili, 2004.
- [7] F. Galton. *Fingerprints*. Macmillan and Co. And New York, 1892.
- [8] A. K. Hrechak and J. A. Mchugh. Automated fingerprint recognition using structural matching. *Pattern Recognition*, 23(8), 1990.
- [9] M. Eleccion. Pattern recognition - automatic fingerprint identification. *IEEE Spectrum*, 1973.
- [10] S. Sharma and S. M. Ranta. Fingertip matching tool final project report. Consultado a 5 de novembro de 2009, 2003.
- [11] S. Patel. <http://fvs.sourceforge.net/about.html>, 2002. Consultado a 10 de Novembro de 2009.

- [12] S. Patel. Fingerprint verification system flowcharts. <http://fvs.sourceforge.net/flowcharts.pdf>. Consultado a 10 de Novembro de 2009.
- [13] C. I. Watson, M. D. Garris, E. Tabassi, C. L. Wilson, M. McCabe, S. Janet, and K. Ko. User's guide to nist biometric image software, 2004. Consultado a 8 de novembro de 2009.
- [14] NIST Open Source Group. Nigos licence. Consultado a 8 de Novembro de 2009.
- [15] NeuroTechnology. Fingerprint identification for pc and web solutions. [http://www.neurotechnology.com/download/VeriFinger\\_SDK\\_Brochure\\_2009-11-02.pdf](http://www.neurotechnology.com/download/VeriFinger_SDK_Brochure_2009-11-02.pdf). Consultado a 10 de Novembro de 2009.
- [16] fingerprint i.t. [http://www.fingerprint-it.com/sol\\_verifinger.html](http://www.fingerprint-it.com/sol_verifinger.html). Consultado a 10 de Novembro de 2009.
- [17] Wikipedia. [http://en.wikipedia.org/wiki/fingerprint\\_verification\\_competition](http://en.wikipedia.org/wiki/fingerprint_verification_competition). Consultado a 7 Novembro de 2009.
- [18] Griaule Biometrics. <http://www.griaulebiometrics.com/page/pt-br/node/334>. Consultado a 7 de Novembro de 2009.
- [19] M. S. Lima. Classificação de impressões digitais via análise de pontos singulares. *XX Brazilian Symposium on Computer Graphics and Image Processing*, 2007.
- [20] S. Patel. Fingertip verification system. <http://fvs.sourceforge.net/fvs.ppt>. Consultado a 10 de Novembro de 2009.
- [21] Wikipedia. <http://en.wikipedia.org/wiki/biometrics>. Consultado a 7 Novembro de 2009.
- [22] S. M. Costa. Classificação e verificação de impressões digitais. Master's thesis, Universidade de São Paulo, 2001.
- [23] C. V. K. Rao. Feature extraction for fingerprint classification. *Pattern Recognition*, 8, 1976.
- [24] B. Moayer and K. S. Fu. Syntatic approach to fingerprint pattern recognition. *Pattern Recognition*, 7, 1975.
- [25] J. L. Blue, G. T. Candela, J. P. Grother, R. Chellapar, and C. L. Wilson. Evaluation of pattern classifiers for fingerprint and ocr applications. *Pattern Recognition*, 27(4), 1994.

- [26] C. L. Wilson, T. G. Candela, and C. I. Watson. Neural network fingerprint classification. *Journal of Artificial Neural Networks*, 1994.
- [27] Q. Xiao and H. Raafat. Fingerprint image postprocessing: A combines statistical and structural approach. *Pattern Recognition*, 24(10), 1991.
- [28] M. Kawagoe and A. Tojo. Fingerprint pattern classification. *Pattern Recognition*, 17(3), 1984.
- [29] K Karu and A. J. Jain. Fingerprint classification. *Pattern Recognition*, 29(3), 1996.
- [30] F. M. Viola. Estudo sobre formas de melhoria na identificação de características relevantes em imagens de impressão digital. Master's thesis, Universidade Federal Fluminense, 2006.
- [31] M. S. Lima. Gaudi-afis: Manual de referência, utilização e desenvolvimento., 2007.
- [32] A. Jain and S. Pankanti. Fingerprint classification and matching. In *HANDBOOK FOR IMAGE AND VIDEO PROCESSING*. Academic Press, 2000.
- [33] A. Farina, Z. M. Koka, and A. Leone. Fingerprint minutiae extraction from skeletonized binary images. *Pattern Recognition*, 32, 1999.
- [34] J. Ravi, K. B. Raja, and K. R. Venugopal. Fingerprint recognition using minutia score matching. *International Journal of Engineering Science and Technology*, 1, 2009.
- [35] F. Zhao and X. Tang. Preprocessing for skeleton-based fingerprint minutiae extraction. *CISST 02 International Conference*, 2002.
- [36] R. Q. Feitosa, F. Carasso, and A. Otoni. Sarid - sistema automático de reconhecimento de impressões digitais.
- [37] C. Barral, J. Coron, and D. Naccache. Externalized fingerprint matching.
- [38] C. M. S. Reis and C. A. R. Pacheco. Autenticação com impressão digital. Master's thesis, Instituto Superior de Engenharia de Lisboa, 2003.
- [39] Sharat S. Chikkerur. Online fingerprint verification system. Master's thesis, State University of New York at Buffalo, 2005.
- [40] J. A. Nalon. Enriquecimento de imagens de impressões digitais. Master's thesis, Universidade Estadual de Campinas, 1997.

- [41] S. L. G. Oliveira, F. Viola, and A. Conci. Filtro adaptativo para melhoria de imagens de impressões digitais utilizando o filtro de gabor e campos direcionais. *4º Congresso Temático de Dinâmica, Controle e Aplicações*, 2005.
- [42] R. Thai. Fingerprint image enhancement and minutiae extraction. Master's thesis, School of Computer Science and Software Engineering, The University of Western Australia, 2003.
- [43] T. S. Castro. Identificação de impressões digitais baseada na extração de minúcias. Master's thesis, Universidade Federal de Juiz de Fora, 2008.
- [44] J. D. (Engº) Breda. Sistema automático de reconhecimento de impressões digitais. Master's thesis, Universidade Estadual de Campinas, 1996.
- [45] Centro Estadual de Pesquisas em Sensoriamento Remoto e Meteorologia. Página dinâmica para aprendizado do sensoriamento remoto. <http://www6.ufrgs.br/engcart/PDASR/maniphist.html>. Consultado a 19 de Maio de 2010.
- [46] V. P. Bastos. Técnicas de segmentação de imagens para recuperação de informações visuais. <http://paginas.ucpel.tche.br/vbastos/>. Consultado a 20 de Maio de 2010.
- [47] R. Fisher, S. Perkins, A. Walker, and E. Wolfart. Hypermedia image processing reference. <http://homepages.inf.ed.ac.uk/rbf/HIPR2/>, 2004. Consultado a 20 de Maio de 2010.
- [48] D. N. Muller and E. L. Daronco. Filtros espaciais passa-baixa. Master's thesis, Universidade Federal do Rio Grande do Sul, 2000.
- [49] C. I. Watson, G. T. Candela, and P. J. Grother. Comparison of fft fingerprint filtering methods for neural network classification. 1994.
- [50] A. K. Jain. *Fundamentals Of Digital Image Processing*. Prentice Hall, 1989.
- [51] L. L. O. Junior. Filtros compostos e adaptativos: o filtro de gaussiano, laplaciano do gaussiano e de gabor (harmônico-gaussiano).
- [52] N. K. Ratha and R. Bolle. *Automatic Fingerprint Recognition Systems*. SpringerVerlag, 2003.
- [53] L. Hong, Y. Wan, and A. Jain. Fingerprint image enhancement: Algorithm and performance evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:777–789, 1998.

- [54] L. P. P. Santos. Visão por computador, 1995. Texto de Apoio.
- [55] D. R. Faria. Reconhecimento de impressões digitais com baixo custo computacional para um sistema de controle de acesso. Master's thesis, Universidade Federal do Paraná, 2005.
- [56] A. Martin and S. Tosunoglu. Image processing techniques for machine vision.
- [57] T. Y. Zhang and C. Y. Suen. A fast parallel algorithm for thinning digital patterns. *Communications of the ACM*, (3), 1984.
- [58] K. Nilsson and J. Bigun. Localization of corresponding points in fingerprints by complex filtering, 2003.
- [59] K. Nilsson and J. Bigun. Complex filters applied to fingerprint images detecting prominent symmetry points used for alignment. In *in Proc. Workshop on Biometric Authentication (in ECCV 2002)*, 2002.
- [60] K. Nilsson and J. Bigun. Prominent symmetry points as landmarks in finger print images for alignment. In *ICPR '02: Proceedings of the 16 th International Conference on Pattern Recognition (ICPR'02) Volume 3*, Washington, DC, USA, 2002. IEEE Computer Society.
- [61] J. Li, W. Yau, and H. Wang. Combining singular points and orientation image information for fingerprint classification. New York, NY, USA, 2008. Elsevier Science Inc.
- [62] B. K. Kumar and K. S. Prasas. A hybrid core point localization algorithm. *IJCSNS International Journal of Computer Science and Network Security*, 2009.
- [63] J. C. Yang and D. S. Park. Fingerprint verification based on invariant moment features and nonlinear bpnn. *International Journal of Control, Automation, and Systems*, 2008.
- [64] C. Mazetti. Metodologias para extracção de características invariantes à rotação em imagens de impressões digitais. Master's thesis, Escola de Engenharia de São Carlos, 2006.
- [65] R. Mazi and A. Júnior. Identificação biométrica através da impressão digital usando redes neurais artificiais. *Anais do XIV ENCITA 2008*, 2008.
- [66] E. Jardini. Mfis: Algoritmo de reconhecimento e indexação em base de dados de impressões digitais em espaço métrico. Master's thesis, Universidade de São Paulo, 2007.





## A. Máscaras para limpeza do esqueleto

Conjunto de todas as máscaras, para remoção de pontos isolados, h-conectados, e com espessura maior do que 1, utilizadas para a limpeza do esqueleto, incluindo as suas rotações.

0	0	0
0	1	0
0	0	0

1	0	1
1	1	1
1	0	1

1	1	1
0	1	0
1	1	1

1	0	0
1	1	0
1	0	0

0	0	1
0	1	1
0	0	1

1	1	1
0	1	0
0	0	0

0	0	0
0	1	0
1	1	1

x	1	0
x	1	1
0	x	x

0	1	x
1	1	x
x	x	0

0	x	x
x	1	1
x	1	0

x	x	0
1	1	x
0	1	x

x	x	x
0	1	0
1	0	1

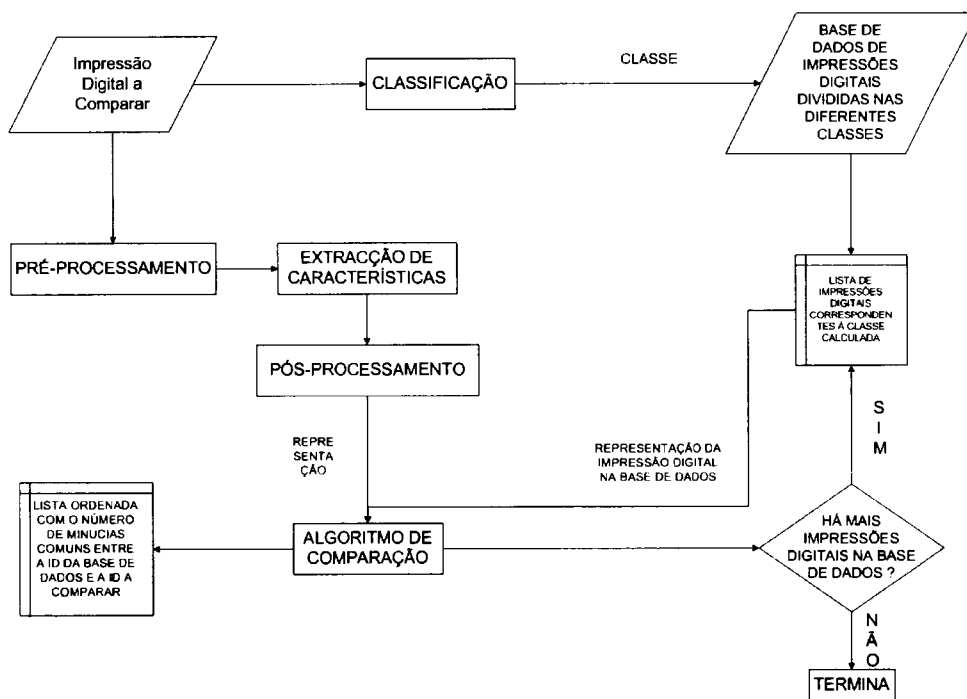
1	0	x
0	1	x
1	0	x

1	0	1
0	1	0
x	x	x



## B. Diagrama do algoritmo de comparação de impressões digitais

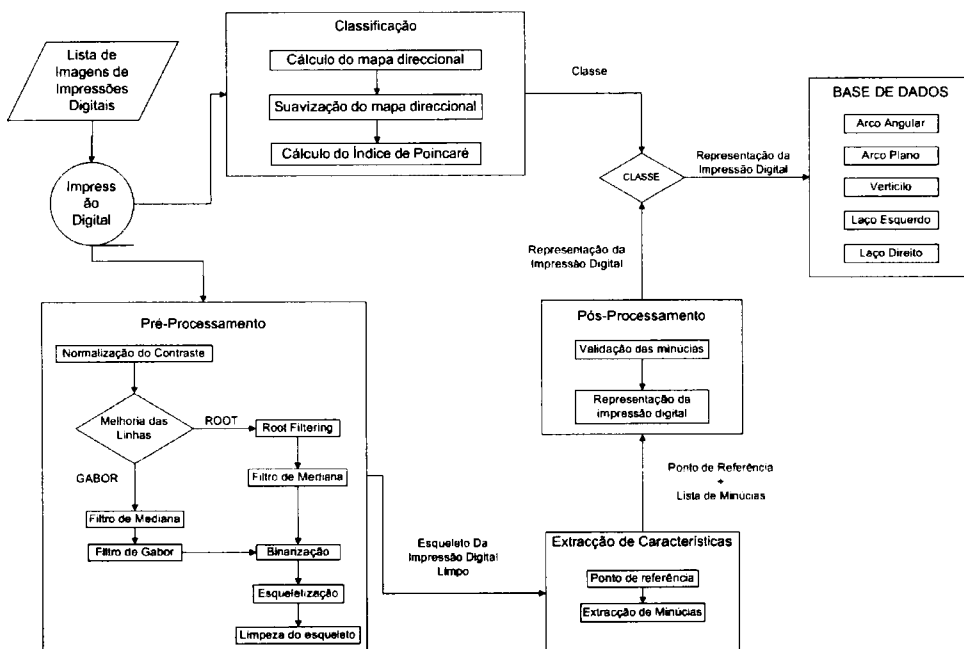
Representação do algoritmo de comparação de impressões digitais, sob a forma de diagrama.





# C. Diagrama do módulo de introdução de impressões digitais na base de dados

Representação, sob a forma de diagrama, do funcionamento do módulo de introdução de impressões digitais, numa base de dados, da aplicação prática desenvolvida.





## D. Código fonte do algoritmo Root Filtering

Implementação em GNU Octave, do algoritmo Root Filtering, utilizado na aplicação prática.

```
function resultado = filtro_fft(imagem, alpha)
imagem = double(imagem);
fourier = fft2(imagem);
magnitudo = abs(fourier);
nova_img = (magnitudo.^alpha).*imagem;
resultado = real(ifft2(nova_img));
```



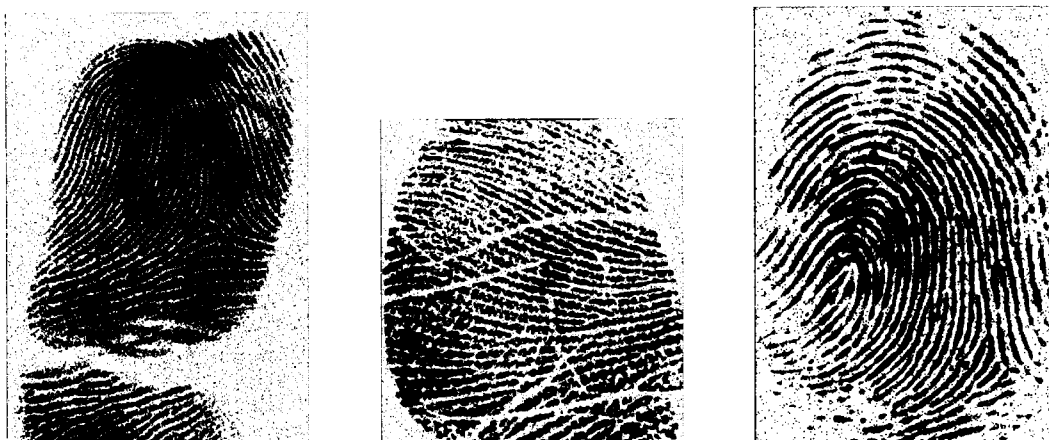


## **E. Exemplos de impressões digitais utilizadas nos testes**

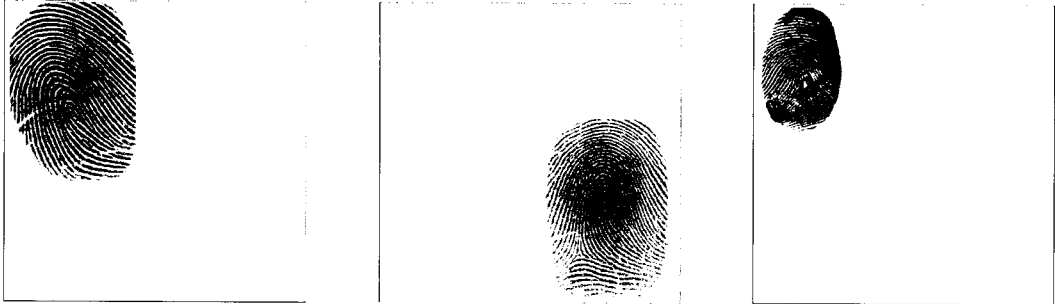
Impressões digitais originais.



Impressões digitais com ruído adicionado.



Impressões digitais deslocadas.



Impressões digitais rotacionadas.



Impressões digitais parciais.



# Índice Remissivo

- aplicações AFIS, 13, 31
  - eFinger, 13, 16
  - F. V. Competition, 15, 19
  - Fingerprint Verification System, 14, 17
  - NIST Biometric Image Software, 14, 18
  - Verifinger, 15, 19
- arcos
  - arco angular, 21
  - arco plano, 21
- aspectos básicos, 8
  - cristas bifurcadas, 8
  - cristas finais, 8
- aspectos compostos, 8
  - cristas curtas, 8, 9
  - cruzamentos, 8, 9
  - esporas, 8, 9
  - ilhas, 8, 9
- binarização, 16, 31, 32, 40, 45
  - binarização adaptativa, 41
- biometria, 10
  - sistemas biométricos, 15
- Bozorth, 18
- classificação, 5, 8, 21, 29, 59
- contraste, 32, 33
- cristas, 25, 32, 34
- cristas papilares, 5, 7
- crossing number, 51, 54
- dactiloscopia, 6
- deformação não-linear, 10
- deslocamento, 9, 10
- direcções, 23
- distância euclidiana, 57
- domínio
  - espacial, 35
  - frequência, 35
- Edward Henry, 21
- esqueletização, 16, 31, 32, 42, 46
  - Zhang-Suen, 42
- filtro complexo, 48
- filtro de Gabor, 37–39
- filtro de mediana, 32, 34
- Francis Galton, 21
- histograma, 32, 33
- histograma uniforme, 32
- identificação, 5, 15, 59
- imutabilidade, 5
- índice de Poincaré, 24, 28, 29
- laços, 21
  - laços direitos, 21, 29
  - laços esquerdos, 21, 29
- lofoscopia, 5, 6
- métodos classificação
  - análise estrutural, 23
  - estatístico, 23



híbridos, 23  
redes neuronais artificiais, 23  
  sintáctico, 23  
magnitude, 35, 50  
mapa direccional, 24, 26–28, 38, 39  
mediana, 34  
moda, 26  
  
onda sinusoidal, 37, 39  
  
padrões, 23, 24  
pelmatoscopia, 6  
perenidade, 5  
ponto de referência, 47, 56  
pontos característicos, 6, 8, 9, 61  
  minúcias, 7, 8, 14, 17, 18, 31  
pontos singulares, 6, 7, 23  
  delta, 8, 21, 29  
  núcleo, 7, 29, 48  
Purkinje, 21  
  
quirosocopia, 6  
  
rotação, 10, 14, 15, 18  
  
sobreposição parcial, 10  
suavização, 24, 26  
  
tangente, 57  
transformada de Fourier, 35  
  
vales, 25, 34  
variabilidade, 5  
verticilos, 22  
  
x-signature, 38, 39