

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA



EXTRAÇÃO DE CARACTERÍSTICAS
DE IMAGENS DE IMPRESSÃO DIGITAL

LUCIANO XAVIER MEDEIROS

AGOSTO

2006

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

EXTRAÇÃO DE CARACTERÍSTICAS DE
IMAGENS DE IMPRESSÃO DIGITAL

Dissertação apresentada por Luciano Xavier Medeiros
à Universidade Federal de Uberlândia para a obtenção do
título de Mestre em Engenharia Elétrica aprovada em
24/08/2006 pela Banca Examinadora:

Professora Edna Lúcia Flôres, Dra (orientadora)

Professor Antonio Cláudio P. Veiga, Dr. (UFU)

Professor Gilberto Arantes Carrijo, Ph.D. (UFU)

Professor Marcos Ferreira de Rezende, Dr. (UNITRI)

Dados Internacionais de Catalogação na Publicação (CIP)

C837o Medeiros, Luciano Xavier.
 Extração de características de imagens de impressão digital / Luciano
Xavier Medeiros. - 2006.
 149f. : il.

 Orientador: Edna Lúcia Flôres.
 Dissertação (mestrado) – Universidade Federal de Uberlândia, Programa de Pós-Graduação em Engenharia Elétrica.
 Inclui bibliografia.

 1. Processamento de imagens - Técnicas digitais - Teses. I. Flôres, Edna Lúcia. II. Universidade Federal de Uberlândia. Programa de Pós-Graduação em Engenharia Elétrica. III. Título.

CDU: 621.397.331

**EXTRAÇÃO DE CARACTERÍSTICAS DE
IMAGENS DE IMPRESSÃO DIGITAL**

LUCIANO XAVIER MEDEIROS

Dissertação apresentada por Luciano Xavier Medeiros à Universidade Federal de Uberlândia como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica.

**Professora Edna Lúcia Flôres, Dra. Eng.
Orientadora**

**Professor Darizon Alves Andrade, Ph.D.
Coordenador do Curso de Pós Graduação**

**Aos meus pais,
meu e irmão e a
minha bonitinha Luciana**

Agradecimentos

À minha orientadora e amiga Edna Lúcia Flôres quem batalhou junto a mim para o término desse trabalho.

A minha família, principalmente meu pai Francisco e minha mãe Dilma, que me apoiaram o todo o tempo, inclusive nos meus momentos de dificuldade.

Ao meu amigo Flávio que suportava as minhas reclamações.

A minha namorada Luciana, que sem o seu amor, sua paciência e sua compreensão eu não teria conseguido vencer mais essa batalha.

Medeiros, Luciano X., Extração de Características de Imagens de Impressão Digital.

Uberlândia, FEELT – UFU, 2006, 149 p.

RESUMO

As impressões digitais são as linhas presentes nos dedos de cada ser humano e essas são únicas para cada um, ou seja, não existem duas pessoas que possuem a mesma forma com que as linhas da impressão se dispõem em seus dedos. Por esse motivo é que ela vem sendo utilizada para a identificação de pessoas para várias finalidades, como por exemplo, o controle de ponto dos funcionários de uma empresa, a identificação de um criminoso e o acesso de pessoas a áreas restritas de extrema segurança. A proposta deste trabalho é obter melhoramentos no algoritmo de extração de características de impressões digitais desenvolvido por Jain e outros [1]. Os objetivos desta dissertação são: melhorar o tempo de processamento no cálculo da orientação de campo, obter uma melhor qualidade das imagens resultantes no processo de binarização e diminuir o número de minúcias espúrias nessas imagens. O cálculo da orientação de campo aperfeiçoado neste trabalho utiliza a propriedade da comutação e resultou em uma redução no tempo de processamento em torno de 90% em relação ao mesmo cálculo que não utiliza essa propriedade. O método de extração das saliências desenvolvido nesta dissertação utiliza o algoritmo DDA, e resulta em imagens de melhor qualidade. Devido a esse melhoramento obtido no processo de binarização e a remoção de falhas nas saliências, o algoritmo de detecção de minúcias implementado neste trabalho encontra uma menor quantidade de minúcias espúrias presentes nas imagens de impressão digital.

Palavras-chaves: minúcias, orientação de campo, saliências, afinamento e impressão digital.

Medeiros, Luciano X., Extraction of Characteristics from fingerprint images.

Uberlândia, FEELT – UFU, 2006, 149 p.

ABSTRACT

The fingerprints are lines present on fingers of each human and they are unique, in other words, there are not two people with the same distribution of lines on their fingers. For this reason, it has been used for identification of people for many purposes, such as to check the entrancy and exit of workers in a company, the identify criminals and to restrict the access of people to restrict areas of extreme security. The proposal of this work is to improve the characteristic extraction algorithm of fingerprint developed by Jain and others. The goal of this work is to accelerate the processing time in orientation field estimation, in order to get a better quality of images, resulting of binarization process and to decrease the number of spurious minutiae on the images. The time processing of orientation field estimation improved in these work uses the commutative propriety is smaller than the same estimation wich does not use this propriety in 90%. The method of ridges extration developed in this work uses the DDA algorithm, and results in better quality images. Due to this improviment got in binarization process and the ridges break removing, the detection minutiae algorithm implemented in this work find a small quantity of spurious minutae present in the fingerprint images.

key-words: minutiae, filed orientation, ridges, thinning and fingerprint.

EXTRAÇÃO DE CARACTERÍSTICAS DE IMAGENS DE IMPRESSÃO DIGITAL

SUMÁRIO

1. Introdução	1
1.1. Introdução	1
1.2. Levantamento Bibliográfico dos Trabalhos Desenvolvidos em Impressão Digital.	8
1.3. Proposta deste Trabalho	12
1.4. Estrutura desta Dissertação	12
1.5. Considerações Finais deste Capítulo	14
 2. Fundamentos em Processamento Digitais de Imagens	 15
2.1. Introdução	15
2.2. Um Modelo Simples de Imagem Digital	15
2.3. Vizinhança de um Pixel	17
2.4. Distância Euclidiana	18
2.5. Filtragem Espacial	19
2.6. Convolução Discreta	21
2.7. Cálculo do Gradiente	23
2.8. Cálculo da Orientação de Campo	27

2.9. Determinação da Orientação de Campo Utilizando o Gradiente	28
2.10. Considerações Finais deste Capítulo	33
3. Métodos de Extração de Características em Imagens de Impressões Digitais	34
3.1. Introdução	34
3.2. A Impressão Digital e suas Características	35
3.3. Cálculo da Orientação de Campo de uma Imagem de Impressão Digital	39
3.3.1. Cálculo do somatório unidimensional utilizando a propriedade da comutação	42
3.3.2. Aplicando a propriedade da comutação no cálculo da orientação de campo	49
3.4. Cálculo do Nível de Consistência da Orientação de Campo	63
3.5. Determinação da Área de Interesse	65
3.6. Extração de Saliências	66
3.6.1. Geração de máscaras de convolução mais suavizadas	73
3.6.2. Algoritmo Digital Differential Analyzer (DDA)	74
3.7. Afinamento das Saliências de Impressões Digitais	77
3.8. Detecção de Minúcias	82
3.9. Correção de falhas presentes em saliências de uma imagem de impressão digital	84
3.10. Determinação das Coordenadas dos Pixels de uma Saliência	89
3.11. Conversão das Saliências em Sinais Discretos Unidimensionais	91
3.12. Cálculo da Distância Média entre as Saliências	105

3.13. Considerações Finais deste Capítulo	108
4. Resultados Obtidos	110
4.1. Introdução	110
4.2. Resultados Obtidos do Cálculo da Orientação de Campo	110
4.3. Desempenho do Cálculo da Orientação de Campo	114
4.4. Resultados Obtidos do Cálculo do Nível de Consistência	116
4.5. Resultados Obtidos na Determinação da Área de Interesse da Imagem	118
4.6. Resultados Obtidos da Extração de Saliências	120
4.7. Afinamento das Saliências	124
4.8. Correção das Falhas Presentes nas Saliências	126
4.9. Resultados Obtidos do Cálculo da Distância Média entre as Saliências	129
4.10. Resultados Obtidos na Detecção de Minúcias	134
4.11. Conclusões	138
5. Conclusões, Contribuições deste Trabalho e Sugestões Para Trabalhos Futuros	140
5.1. Conclusões	140
5.2. Contribuições Deste Trabalho	142
5.3. Sugestões Para Trabalhos Futuros	144
Referência Bibliográficas	147

LISTA DE FIGURAS

1.1 –	Valores de TRF, TRI e TIE para três tipos de aplicações	4
2.1 –	Convenção dos eixos utilizada para representar as imagens digitais neste trabalho.	16
2.2 –	Vizinhança de um pixel de coordenada (x, y)	17
2.3 –	Representação dos pixels p e q e a distância Euclidiana D_e no plano cartesiano	18
2.4 –	Máscara w de tamanho 3×3	20
2.5 –	Filtragem de $f(x,y)$ por uma máscara w	21
2.6 –	(a) Sinal original; (b) sinal rebatido e translado ($x = y = 0$)	23
2.7 –	Operadores utilizados para o cálculo do gradiente em imagens. Aplicação dos operadores de Sobel. (a) imagem original; (b) componente G_x ; (c) componente G_y ; (d) gradiente da imagem original.	25
2.8 –	Aplicação dos operadores de Sobel. (a) imagem original; (b) componente G_x ; (c) componente G_y ; (d) gradiente da imagem original	26
2.9 –	Segmentos de retas em uma região de uma imagem	27
2.10 –	(a) Segmento de reta AB; (b) direção θ considerando A como início do vetor; (c) direção $\theta + \pi$ considerando B como início do vetor.	28
2.11 –	(a) Representação do vetor no plano xy ; (b) componentes G_x e G_y representadas nesse plano.	29
3.1 –	Região de uma impressão digital	35
3.2 –	Tipos de impressão digital. (a) arco; (b) arco “tented”; (c) loop à esquerda; (d) loop à direita; (e) “whorl”; (f) loop gêmeos	36

3.3 –	Tipos de minúcias. (a) terminação de saliência; (b) bifurcação de saliência	37
3.4 –	Imagens de uma mesma impressão obtidas por um scanner. (a) imagem com uma pressão maior e mais distribuída do que a da letra (b).	38
3.5 –	Imagens da mesma impressão, a letra (b) está translada e rotacionada em relação a letra (a).	38
3.6 –	Imagem dividida em quatro blocos	40
3.7 –	A não propagação do ruído em outros blocos da imagem	41
3.8 –	Pixels afetados pelo ruído presente no bloco superior-esquerdo	42
3.9 –	Sinal discreto $f(x)$	43
3.10 –	Sinal discreto $g(x)$ de comprimento $L = 18$	44
3.11 –	Elementos pertencentes. (a) $g(7)$; (b) $g(8)$; (c) $g(7)$ e $g(8)$	45
3.12 –	Fluxograma do algoritmo aperfeiçoado neste trabalho para o cálculo da orientação e campo usando a propriedade da comutação	54
3.13 –	(a) Valores de $f(i, j)$, (b) resultado da Equação (3.28)	56
3.14 –	Valores de V_{xs} para as linhas. (a) $i = 0$; (b) $i = 1$; (c) $i = 2$; (d) $i = 3$; (e) $i = 4$.	57
3.15 –	Coordenadas de uma máscara de convolução 11×7 utilizada neste trabalho	69
3.16 –	Resultado da máscara h_t da Figura 3.16	72
3.17 –	Máscara h_t de tamanho 11×7 , $\alpha = -60^\circ$ e $\delta = 25.000$	72
3.18 –	Ângulo α formado entre a reta da Equação (3.66) e o eixo x	75
3.19 –	Vizinhança de oito do pixel p_i	77
3.20 –	Vizinhança de oito, onde $N(p_i) = 4$ e $S(p_i) = 3$	78
3.21 –	Afinamento de uma imagem. (a) imagem original; (b) pixels eliminados pelo 1º passo; (c) pixels eliminados pelo 2º passo; (d) imagem resultante do	80

afinamento com os pixels eliminados	
3.22 – (a) Imagem original; (b) imagem afinada pelo o algoritmo de afinamento	81
3.23 – Vizinhanças de oito dos pixels que geram falsas minúcias	82
3.24 – (a) Minúcia do tipo terminação; (b) minúcia do tipo bifurcação	82
3.25 – Exemplo de duas falsas minúcias	83
3.26 – Vizinhanças de oito para minúcias do tipo bifurcação com $N(p_1) = 3$	83
3.27 – Fluxograma do algoritmo para eliminação de falhas em saliência	86
3.28 – Exemplo de eliminação de uma falha em uma saliência	87
3.29 – Teste do próximo pixel	90
3.30 – Saliência com a minúcia de coordenada (4, 2)	94
3.31 – Exemplo de uma função contínua em um intervalo [a, b]	96
3.32 – Sinal unidimensional resultante da saliência mostrada na Figura 3.30	98
3.33 – Direção de uma minúcia do tipo bifurcação	99
3.34 – Algoritmo de determinação da saliência que será associada a uma minúcia do tipo bifurcação	102
3.35 – Passos do algoritmo de determinação da saliência que será associada a uma minúcia do tipo bifurcação. (a) região 10×10 de uma minúcia; (b) três possíveis pixels; (c) extremos de cada saliência; (d) saliência escolhida na cor cinza.	103
3.36 – Região onde será calculada a distância média entre as saliências	106
3.37 – Máscara de convolução 7×7	106
4.1 – Exemplo do cálculo da orientação de campo. (a) imagem original; (b) visualização da orientação de campo	112
4.2 – Exemplo do cálculo da orientação de campo. (a) imagem original; (b)	113

visualização da orientação de campo	
4.3 – Gráfico do tempo gasto no cálculo da orientação de campo para imagens de tamanho $N \times N$ pixels.	115
4.4 – Exemplo de re-estimação da orientação de campo. (a) orientação de campo original; (b) orientação de campo re-estimada	117
4.5 – Determinação da área de interesse. (a) imagem original; (b) área de interesse com resíduos; (c) área de interesse melhorada pela solução implementada neste trabalho	119
4.6 – Resultado da detecção de saliências das máscaras de convolução geradas sem a utilização do algoritmo DDA. (a) imagem original; (b) e (c) saliências detectadas pelas máscaras h_t e h_b , respectivamente; (d) resultado final da detecção de saliências	121
4.7 – Resultado da extração das saliências de máscaras de convolução geradas com a utilização do algoritmo DDA. (a) imagem original; (b) e (c) saliências extraídas de h_t e h_b , respectivamente; (d) resultado final da extração de saliências	123
4.8 – Correção do afinamento das saliências. (a) imagem resultante do algoritmo de afinamento; (b) correção do afinamento da imagem da letra (a)	124
4.9 – Resultado do algoritmo de afinamento aplicado a imagem da letra (a); (b) imagem resultante do afinamento; (c) melhoramento neste trabalho do imagem da letra (b)	125
4.10 – Exemplo de correção das falhas nas saliências. (a) imagem binarizada de uma impressão digital. (b) imagem resultante após a correção das falhas na imagem da letra (a)	127
4.11 – Exemplo de correção das falhas nas saliências. (a) imagem binarizada de uma impressão digital. (b) imagem resultante após a correção das falhas na imagem da letra (a)	128

4.12 –	Imagens da impressão digital 1 utilizadas para cálculo da distância: (a) d_1 ; (b) d_2 ; (c) d_3 ; (d) d_4 ; (e) d_5 ; (f) d_6 .	130
4.13 –	Imagens da impressão digital 2 utilizadas para cálculo da distância: (a) d_1 ; (b) d_2 ; (c) d_3 ; (d) d_4 ; (e) d_5 ; (f) d_6 .	131
4.14 –	Imagens da impressão digital 3 utilizadas para cálculo da distância: (a) d_1 ; (b) d_2 ; (c) d_3 ; (d) d_4 ; (e) d_5 ; (f) d_6 .	132
4.15 –	Falha em uma região de uma imagem de impressão digital em níveis de cinza	133
4.16 –	Detecção de falsas minúcias. (a) resultado do algoritmo de afinamento; (b) resultado do algoritmo de detecção de minúcias aplicado na Figura 4.16(a)	134
4.17 –	Exemplo de detecção de minúcias. (a) imagem binarizada; (b) e (c) resultados da detecção de minúcias sem e com remoção das falhas nas saliências, respectivamente	136
4.18 –	Minúcias espúrias detectada na região central da Figura 4.13(b)	137

LISTA DE TABELAS

Tabela 1.1 – Valores de TRF, TRI e TIE para três tipos de aplicações	4
Tabela 1.2 – Recursos computacionais utilizados neste trabalho	12
Tabela 3.1 – Número de adições com e sem utilizar a propriedade da comutação	48
Tabela 3.2 – Número de adições para as somatórias bidimensionais considerando $W = 16$	63
Tabela 3.3 – Valores de u a partir de v no algoritmo para a geração das máscaras de convolução h_t e h_b .	71
<i>Tabela 3.4 – Coordenadas dos pixels vizinhos ao pixel p_1 da Figura 3.29</i>	90
Tabela 3.5 – Coordenadas originais e resultantes da translação e da rotação dos pixels pertencentes a saliência da Figura 3.30.	95
<i>Tabela 3.6 – Novas coordenadas arredondadas dos pixels da Figura 3.30</i>	98
<i>Tabela 4.1 – Tempo gasto no cálculo da orientação de campo com e sem a utilização da propriedade da comutação.</i>	114
<i>Tabela 4.2 – Distâncias médias entre as saliências para três impressões digitais diferentes.</i>	129

CAPÍTULO I

INTRODUÇÃO

1.1 – Introdução

Atualmente, as pessoas estão cada vez mais interessadas em sistemas de segurança. A finalidade desses sistemas são permitir acessos de pessoas a áreas restritas e proteger bens como: dinheiro, jóias, documentos importantes, arquivos de computadores, casa, carro, etc.. Com a diminuição dos preços e o aumento na velocidade de processamento dos computadores, está se tornando comum a utilização de *biometria* na área de segurança.

A *biometria* é a medição das características físicas do ser humano. Algumas dessas características são utilizadas na identificação de uma pessoa, como por exemplo, reconhecimento utilizando imagens digitais da íris, impressão digital, impressão palmar, etc..

Existem dois sistemas de segurança que podem utilizar biometria. Esses sistemas são: *autentificação* e *identificação*.

A *autentificação* consiste em verificar se uma pessoa é ela mesma por uma característica biométrica (impressão digital, íris, fala) e por um outro dado de entrada, como por exemplo, cartões de identificação, nome do usuário (acesso a computadores), senhas, etc.. A vantagem da autentificação é realizar uma procura em um banco de dados de um para um, ou seja, a comparação é realizada entre a imagem de entrada e entre uma imagem armazenada

no banco de dados. Por exemplo, uma pessoa que trabalha em uma empresa, onde o sistema de controle de entrada e saída é realizado por um computador através de um cartão de acesso juntamente com a impressão digital. Esse computador compara a impressão digital de entrada com a impressão digital armazenada no banco de dados referente ao nome do funcionário do cartão. Se a impressão digital é a mesma, o acesso é permitido, caso contrário o acesso é negado. Nesse exemplo, houve a comparação entre duas imagens.

O sistema de *identificação* consiste em identificar uma pessoa por uma característica biométrica, por exemplo, dizer quem é o dono da imagem de impressão digital de entrada. Esse sistema tem a desvantagem de realizar uma comparação de 1 para N , onde N é o número de imagens armazenadas no banco de dados. Por exemplo, se uma empresa utiliza um sistema de identificação para controle de entrada e saída pelo reconhecimento da impressão digital de um funcionário, o computador pode chegar a realizar uma comparação da imagem de entrada com todas as imagens armazenadas no banco de dados.

Para um sistema que utiliza a autenticação, o tempo de processamento é pequeno, pois trata-se somente de uma possibilidade de acesso. Já nos sistemas de identificação, o tempo de processamento pode ser muito longo, causando desconforto e longas filas (no caso de acesso a um local público) dependendo do número de imagens armazenadas no banco de dados. Algumas técnicas estatísticas, como por exemplo técnicas baseadas na frequência de utilização podem ser aplicadas para diminuir o tempo de procura do sistema de identificação. Uma outra forma de diminuir esse tempo de procura do sistema é pela classificação do usuário pelo tipo de impressão digital, restringindo a procura somente ao grupo da pessoa.

Nos processos de identificação ou autenticação, podem ocorrer *erros de reconhecimento*. Estes erros são representados por duas taxas, a *Taxa de Falso Reconhecimento* (TFR) e a *Taxa de Reconhecimento Incorreto* (TRI).

Para duas imagens de impressão digital, o sistema de reconhecimento calcula o grau de similaridade (s). Esse sistema compara esse valor com um limiar t . Caso s é maior do que t , as duas imagens são da mesma impressão digital, caso contrário, elas são imagens de impressões digitais diferentes. O valor de t é determinado empiricamente a partir de testes com um banco de dados de imagens, pois o valor de t depende do algoritmo de reconhecimento utilizado.

A *Taxa de Falso Reconhecimento* (TFR) é a probabilidade de uma pessoa cadastrada em um banco de dados não ser identificada corretamente, ou seja, o sistema de identificação reconhece esta pessoa como sendo outra. A *Taxa de Reconhecimento Incorreto* (TRI) é a probabilidade de uma pessoa não cadastrada em um banco de dados ser identificada como cadastrada nesse banco.

A Figura 1.1 mostra a relação entre a TFR e a TRI para três tipos de aplicações de reconhecimento. Esses tipos são: aplicações de investigação policiais, aplicações de uso civil (controle de ponto em uma empresa, controle de acesso de usuário de um clube, etc.) e aplicações de acesso de alta segurança.

Na Figura 1.1 a Taxa de Igualdade de Erro (TIE) indica a taxa de erro para o limiar t onde para as taxas TFR e TRI são idênticas. Embora a TIE é um indicador importante, na prática um sistema de reconhecimento de impressão digital raramente utiliza esse indicador. Nesse sistema utiliza-se um limiar mais preciso para reduzir a TFR apesar do crescimento da TRI, como mostrado nessa figura.

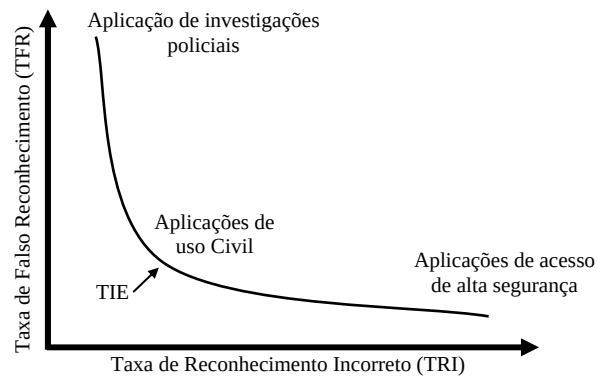


Figura 1.1 – Valores de TRF, TRI e TIE para três tipos de aplicações.

Como mostrado por Jain e outros [1], uma característica biométrica para ser utilizada na autenticação ou na identificação de uma pessoa, deve satisfazer os seguintes requerimentos:

- *universalidade*, ou seja, todas as pessoas deve possuir essa característica;
- *exclusividade*, indica a diferença de uma mesma característica biométrica relativo a duas pessoas;
- *permanência*, significa que a característica não desaparece com o passar do tempo;
- *coletividade*, ou seja, a característica pode ser medida quantitativamente;
- *desempenho*, indica quanto o sistema é preciso na identificação de uma pessoa, ou seja, identificação correta ou errônea de uma pessoa;

- *aceitabilidade*, indica o quanto as pessoas aceitam um sistema biométrico, ou seja, quanto maior a aceitabilidade, menor é a relutância das pessoas em utilizar um sistema biométrico; e
- *Dificuldade de falsificação*, ou seja, é a facilidade de enganar o sistema que utiliza a característica biométrica por técnicas fraudulentas.

A Tabela 1.1 [1] mostra a comparação entre algumas tecnologias biométricas. Essa comparação é realizada pelos os requerimentos citados anteriormente.

Tabela 1.1 – Comparação entre tecnologias biométricas.

<i>Tecnologia Biométrica</i>	<i>Universalidade</i>	<i>Exclusividade</i>	<i>Permanência</i>	<i>Coletividade</i>	<i>Desempenho</i>	<i>Aceitabilidade</i>	<i>Dificuldade falsificação</i>
Face	alto	baixo	médio	alto	baixo	alto	baixo
Impressão digital	médio	alto	alto	médio	alto	médio	alto
Geometria da mão	médio	médio	médio	alto	médio	médio	médio
Veias da mão	médio	médio	médio	médio	médio	médio	alto
Íris	alto	alto	alto	médio	alto	baixo	alto
Retina	alto	alto	médio	baixo	alto	baixo	alto
Assinatura	baixo	baixo	baixo	alto	baixo	alto	baixo
Voz	médio	baixo	baixo	médio	baixo	alto	baixo

A Tabela 1.1 compara os requerimentos das tecnologias biométricas utilizando três níveis: baixo, médio e alto.

De acordo com a Tabela 1.1, uma pessoa escolhe um sistema de reconhecimento de acordo os índices dos requerimentos de cada tecnologia biométrica levando em consideração, principalmente, os possíveis erros de reconhecimento (requerimento de *desempenho*), custo

para a implementação do sistema (requerimento *coletividade*) e a segurança do sistema (representado na tabela pelo o requerimento *dificuldade de falsificação*). Este trabalho utiliza a na característica biométrica *impressão digital*.

De acordo com o National Institute of Standard and Technology (NIST) quando é necessário armazenar uma impressão digital em um banco de dados a representação padrão dessa impressão é baseada nas minúcias, incluindo suas localizações e orientações. Nas representações baseadas nas minúcias pode-se incluir um ou mais atributos globais, como a orientação do dedo, localização do núcleo, o delta (pontos especiais de uma impressão digital) e o tipo de impressão digital.

O reconhecimento de impressões digitais é entre as técnicas de autentificação ou identificação de uma pessoa, uma das mais conhecidas e utilizadas. Algumas das atuais aplicações que pode-se empregar esse tipo de reconhecimento são:

- Na partida e na regulagem de equipamentos do carro como: altura e inclinação do banco do motorista, regulagem dos retrovisores e altura do volante. O carro somente aceita esses comando, quando reconhece a impressão digital da pessoa cadastrada em seu banco de dados. Por exemplo, o carro fabricado pela Audi modelo A8 aceita até quatro impressões digitais diferentes em seu banco de dados;
- Sistemas de acessos de entrada de alunos em uma escola, em condomínios residenciais, clubes e empresas de telefonia. Por exemplo, a Telefônica da cidade de São Paulo, permite o acesso aos seus computadores centrais somente pelo reconhecimento da impressão digital e da íris do usuário; e

- Identificação de suspeito de crime pela sua impressão digital que foi coletada na cena do crime. Dependendo da quantidade de impressões digitais armazenadas no banco de dados, o processo de identificação pode levar muito tempo.

As primeiras publicações sobre impressões digitais foram escritas pelo inglês Nehemiah Grew em 1684, onde ele estudou as saliências e as estruturas dos poros em uma impressão digital. Depois dessas publicações, outros estudos foram realizados por Mayer em 1788 que apresentou estudos detalhados sobre a formação anatômica das impressões digitais, por Purkinje em 1823 que propôs o primeiro esquema de classificação de impressões digitais e por Henry Fauld em 1880 que sugeriu a individualidade das impressões digitais baseadas em estudos empíricos.

No início dos anos 60, o Federal Bureau Investigation (FBI) e o departamento de polícia de Paris investiram no desenvolvimento de *sistemas automáticos de identificação de impressões digitais* (Automatic Fingerprint-Identification Systems – AFIS). O objetivo desses sistemas é realizar a identificação de uma pessoa em um tempo menor e mais eficiente do que o realizado por uma pessoa. O AFIS utiliza as características da impressão digital para identificar uma pessoa. Essas características dependem do algoritmo de reconhecimento, como por exemplo, utiliza-se o diagrama de minúcias ou diagrama de saliências entre as imagens. O desempenho do AFIS depende do algoritmo utilizado para o reconhecimento, da qualidade e da quantidade de imagens utilizadas, pois quanto maior essa quantidade, maior é o tempo de processamento.

Este capítulo apresenta um levantamento bibliográfico dos trabalhos desenvolvidos em impressão digital, a proposta deste trabalho e a estrutura desta dissertação. Finalmente são realizadas considerações finais sobre este capítulo.

1.2 – Levantamento Bibliográfico dos Trabalhos Desenvolvidos em Impressão Digital.

Jain e outros [1] desenvolveram um sistema de reconhecimento de impressões digitais, dividido basicamente em quatro etapas: cálculo da orientação de campo calculada a partir do gradiente da imagem, extração das saliências determinada a partir das direções obtidas nesse cálculo da orientação, detecção das minúcias e o reconhecimento de impressões digitais baseado no alinhamento das minúcias.

Verma e Majumdar [2] implementaram o melhoramento e a binarização de imagens de impressão digital. O processo de melhoramento da imagem foi realizado por eles utilizando-se lógica fuzzy na determinação do valor de cada pixel da imagem. Depois do melhoramento a imagem foi dividida em blocos, onde os pixels de cada bloco foram binarizados separadamente do resto da imagem aplicando equações pré-definidas por eles.

Sherlock e Monro [3] desenvolveram uma técnica de enriquecimento da qualidade de uma imagem de impressão digital. Eles utilizaram a técnica de filtragem no domínio da frequência, porque no domínio espacial as máscaras de convolução não podiam ser de dimensões muito grandes, pois computacionalmente não era eficiente. Essa técnica possuía uma estágio de filtragem seguido por um estágio de limiarização (binarização da imagem).

Xiao e Raapat [4] implementaram uma técnica de eliminação de minúcias espúrias do tipo bifurcação e terminação.

O’Gorman e Nickerson [5] desenvolveram uma técnica de filtragem espacial para melhorar imagens de impressão digital utilizando uma máscara de convolução. Esta máscara possui as linhas horizontais e paralelas as saliências e as colunas são verticais e perpendiculares as saliências. Os coeficientes da máscara de convolução são determinados a partir das posições desses coeficientes aplicados a funções trigonométricas. Eles também desenvolveram o cálculo da orientação de campo dividindo a imagem em blocos e calculando essa orientação para o pixel central do bloco a partir dos valores de níveis de cinza da imagem.

Hung [6] implementou um algoritmo de melhoramento de imagem de impressão digital considerando que as saliências e os vales dessa imagem possuem larguras similares e igualmente espaçadas. Para as saliências em que suas larguras não são muito uniformes, esse algoritmo faz uma pequena correção a partir da linha central dessas saliências.

Onnia e Tico [7] desenvolveram o cálculo da distância média entre as saliências de uma imagem digital a partir de suas imagens em nível de cinza. Para a realização desse cálculo foi necessário a determinação da orientação de campo da imagem. Eles dividiram a imagem da impressão digital em blocos e calcularam a distância média de cada bloco. Depois eles calcularam a média das distâncias médias de cada bloco, obtendo assim a distância média da imagem inteira.

Homem e Costa [8] implementaram um algoritmo de afinamento utilizando campo vetorial. Eles calcularam os valores da transformada de Fourier bidimensional desse campo e a partir do resultado desse cálculo eles encontraram duas máscaras de convolução utilizadas no processo de afinamento.

Tong e Tang [9] desenvolveram um algoritmo de casamento de minúcias, onde as minúcias são expressas como números complexos. Para duas imagens de impressão digital que serão comparadas, esse algoritmo gera dois conjuntos de minúcias, onde cada elemento desse conjunto contém as informações de cada minúcia que são: abscissa, ordenada e direção da saliência. O algoritmo de Tong e Tang [9], quando encontra um par de minúcias correspondentes entre as duas imagens, realiza o alinhamento dos diagramas de minúcias e aplica o casamento de minúcias. O algoritmo deles realiza esse casamento tentando verificar a semelhança entre esses diagramas das duas imagens. O casamento de minúcias é realizado contando os pontos coincidentes dessas duas imagens com as coordenadas (x, y) admitindo uma tolerância limitada por um círculo de raio T_{dir} .

Hao, Tan e Wang [10] implementaram um algoritmo de casamento de minúcias entre duas imagens de impressão digital, comparando as minúcias do tipo terminação das duas imagens e depois as minúcias do tipo bifurcação. Para comparar duas minúcias do tipo terminação, eles aplicaram uma equação que define a “similaridade” entre as minúcias, a partir das coordenadas dos pixels das suas respectivas saliências. Para as minúcias do tipo bifurcação também eles utilizaram a “similaridade”, mas a equação considera o valor das três saliências associadas a cada minúcia.

Munir e Javed [11] desenvolveram um algoritmo de reconhecimento de impressão digital que utiliza o diagrama de saliências de cada imagem. A partir desse diagrama foi possível comparar as duas imagens, pois esse diagrama em uma imagem de impressão pode ser considerado como um diagrama de textura orientado tentando sua orientação e frequência espacial dominante em uma vizinhança local. A frequência é devido a distância entre as saliências em uma impressão digital e a orientação é devido ao diagrama de fluxo das

saliências. Extraídas a frequência e a orientação das saliências, uma representação distinta das saliências pode ser encontrada.

Lindoso, Entrena e Liu [12] implementaram o casamento entre duas imagens de impressões digitais baseado na correlação entre essas imagens. A correlação indica a similaridade entre as duas imagens de impressão digital. No cálculo da correlação, também foram considerados a diferença de brilho entre essas imagens, a pressão exercida sobre o scanner no momento da aquisição, as doenças de peles e a translação e a rotação.

Ko [13] desenvolveu um melhoramento das imagens de impressões digitais utilizando análise espectral dessas imagens, ou seja, o melhoramento é realizado no domínio da frequência. Aplicando a transformada de Fourier na imagem de impressão digital, obtém-se o espectro dessa imagem onde vários tipos de filtros são aplicados para obter uma melhora nesse espectro. Depois que todos os filtros são aplicados, utiliza-se a transformada inversa de Fourier para obter a imagem da impressão digital melhorada.

Lee, Wang e Wu [14] implementaram o algoritmo que determina as características de uma imagem de impressão digital no domínio da frequência ao invés de utilizar o domínio espacial. De acordo com as características dos ângulos e as distâncias entre os picos em relação à origem do sistema de coordenadas no domínio da frequência, eles determinam a direção e a distância entre as saliências. Esse algoritmo também utilizou o filtro de Gabor para melhorar a imagem da impressão digital.

1.3 – Proposta deste Trabalho

Este trabalho propõe melhoramentos no algoritmo de extração de características de impressões digitais desenvolvido por Jain e outros [1]. Essas características, dependendo do processo desenvolvido, são utilizadas em algumas fases do reconhecimento de impressões digitais, tais como binarização e detecção de minúcias. Os objetivos desta dissertação são: melhorar o tempo de processamento no cálculo da orientação de campo, obter uma melhor qualidade das imagens resultantes no processo de binarização e diminuir o número de minúcias espúrias nessas imagens.

A Tabela 1.2 mostra os recursos computacionais utilizados neste trabalho.

Tabela 1.2 – Recursos computacionais utilizados neste trabalho

<i>Processador</i>	AMD Athlon XP +2100, 1,7 GHz
<i>Memória RAM (DDR333)</i>	256 MB
<i>Disco Rígido</i>	FUJITSU de 80 GB, 7200 rpm
<i>Ferramenta de programação</i>	Matlab 6.1

1.4 – Estrutura desta Dissertação

Este capítulo apresenta um levantamento bibliográfico dos trabalhos desenvolvidos em impressão digital, a proposta deste trabalho e a estrutura desta dissertação. Finalmente, são realizados considerações finais sobre este capítulo.

O capítulo 2 descreve um modelo simples de imagem digital, vizinhança de um pixel, distância Euclidiana, filtragem espacial, convolução discreta, cálculo do gradiente e

determinação da orientação de campo de uma imagem. Finalmente, são realizados considerações finais sobre esse capítulo.

O capítulo 3 mostra a impressão digital e suas características, o cálculo da orientação de campo para cada pixel da imagem de impressão digital, o cálculo do nível de consistências dessa orientação e a determinação da área de interesse dessa imagem. São descritas as seguintes operações em relação as saliências: extração, afinamento e remoção das falhas. Depois são apresentadas a detecção de minúcias, a correção de falhas presentes em saliências de uma impressão digital, a determinação das coordenadas dos pixels de uma saliência, a remoção e a conversão das saliências em sinais discretos unidimensionais e o cálculo da distância média entre as saliências. Finalmente, são realizados considerações finais sobre esse capítulo.

O capítulo 4 mostra os resultados obtidos na extração das seguintes características de imagens de impressão digital: cálculo da orientação de campo; cálculo do nível de consistência das direções encontradas no cálculo da orientação de campo, o desempenho desse cálculo com e sem a utilização da propriedade da comutação, determinação da área de interesse nessas imagens; detecção, afinamento e correção das falhas presentes nas saliências; cálculo da distância média entre as saliências e a detecção das minúcias nessas saliências. Finalmente, são realizadas conclusões sobre esses resultados.

O capítulo 5 apresenta as conclusões, as contribuições desta dissertação e as sugestões para trabalhos futuros.

1.5 – Considerações Finais deste Capítulo

Este capítulo apresentou um levantamento bibliográfico dos trabalhos desenvolvidos em impressão digital, a proposta deste trabalho e a estruturação desta dissertação.

O próximo capítulo descreve um modelo simples de imagem digital, vizinhança de um pixel, distância Euclidiana, filtragem espacial, convolução discreta, cálculo do gradiente, e determinação da orientação de campo de uma imagem.

CAPÍTULO II

FUNDAMENTOS EM PROCESSAMENTO

DIGITAL DE IMAGENS

2.1 – Introdução

O objetivo deste capítulo é apresentar alguns conceitos fundamentais em processamento digital de imagem para a extração das características necessárias a serem utilizadas no reconhecimento de impressões digitais.

Este capítulo descreve um modelo simples de imagem digital, vizinhança de um pixel, distância Euclidiana, filtragem espacial, convolução discreta, cálculo do gradiente, e determinação da orientação de campo de uma imagem. Finalmente, são realizadas considerações finais deste capítulo.

2.2 – Um Modelo Simples de Imagem Digital

Gonzalez e Woods [15] definiram o termo imagem monocromática ou simplesmente imagem, como uma função bidimensional, indicada por $f(x, y)$, em que o valor ou a amplitude de f na coordenada espacial (x, y) fornece a intensidade de luz (brilho) da imagem naquele ponto. A Figura 2.1 mostra a convenção dos eixos utilizada para representar as imagens digitais neste trabalho.

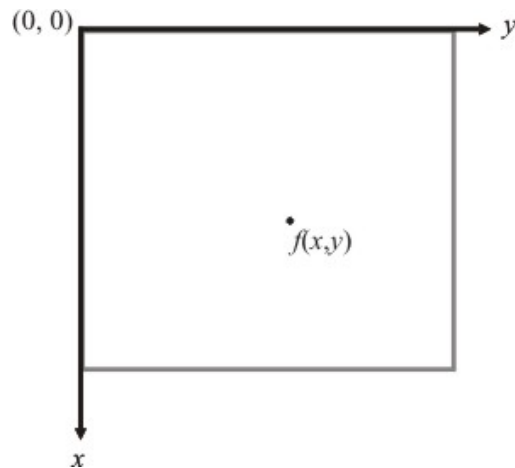


Figura 2.1 – Convenção dos eixos utilizada para representar as imagens digitais neste trabalho.

Uma imagem digital é uma imagem $f(x, y)$ discretizada em coordenadas espaciais e em brilho, $f(x, y)$ pode ser representada por uma matriz, onde os índices das linhas e das colunas indicam um ponto na imagem e o valor do elemento da matriz identifica o nível de cinza ou a cor naquele ponto. Os elementos dessa matriz são chamados pixels (abreviatura de *picture elements*). O tamanho da matriz que representa uma imagem e o número de níveis de cinza variam dependendo da aplicação. É comum utilizar em processamento digital de imagem valores que são potência inteira de dois, isto é [15]:

$$M = 2^p, \quad N = 2^q \quad \text{e} \quad C = 2^k \quad (2.1)$$

onde:

M – número de linhas da imagem.

N – número de colunas da imagem.

C – número de níveis de cinza da imagem.

O número de bits (b) necessários para armazenar uma imagem digital é obtido pela Equação (2.2).

$$b = M \times N \times k \quad (2.2)$$

onde:

k – número de bits necessários para representar um pixel.

Por exemplo, uma imagem 128×128 pixels com 64 níveis de cinza necessita de 98.304 bits para armazená-la.

2.3 – Vizinhança de um Pixel

Considere um pixel de coordenada (x, y) . A vizinhança desse pixel são os outros pixels que estão ao seu redor, como mostrado na Figura 2.2.

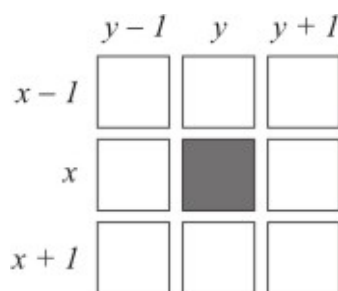


Figura 2.2 – Vizinhança de um pixel de coordenada (x, y) .

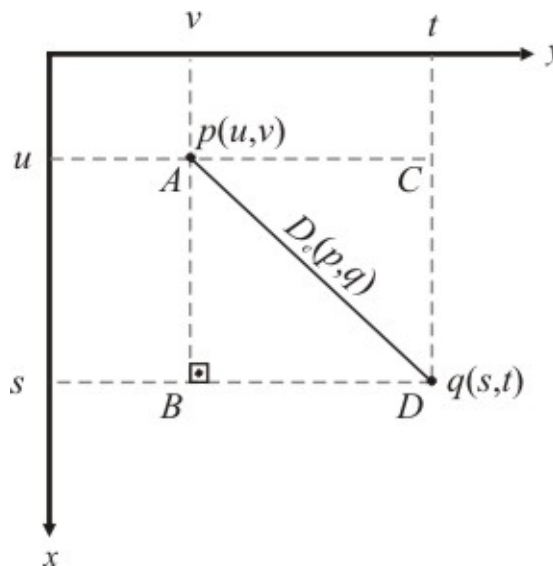
Na Figura 2.2 o pixel central, que possui um tom mais acinzentado em relação aos demais pixels, possui oito vizinhos. Essa figura mostra também as coordenadas de cada pixel, por

exemplo, a coordenada do pixel à direita do pixel central é $(x, y+1)$ em relação a coordenada (x, y) . Se a coordenada do pixel central é, por exemplo, $(7, 6)$, a coordenada do pixel à direita é $(7, 7)$.

Existem pixels que não possuem oito vizinhos. Quando uma vizinhança tem menos do que oito vizinhos ela é chamada de *vizinhança parcial*. Pixels que possuem vizinhanças parciais são os pixels que encontram-se na primeira e na última colunas, isso também ocorre com os pixels localizados na primeira e na última linhas da imagem.

2.4 – Distância Euclidiana

A *distância Euclidiana* é a distância entre dois pixels localizados no plano cartesiano. Considere dois pixels, p e q com coordenadas (u, v) e (s, t) , respectivamente. A Figura 2.3 mostra esses pixels e a distância Euclidiana (D_e) no plano cartesiano.



2.3 – Representação dos pixels p e q e a distância Euclidiana D_e no plano cartesiano

As Equações (2.3) e (2.4) calculam os comprimentos dos lados AB e BD , respectivamente, na Figura 2.3.

$$AB = s - u \quad (2.3)$$

$$BD = t - v \quad (2.4)$$

Utilizando-se o teorema de Pitágoras no triângulo retângulo ABD da Figura 2.3, obtém-se a Equação (2.5).

$$D_e(p, q)^2 = AB^2 + BD^2 \quad (2.5)$$

Substituindo-se as Equações (2.3) e (2.4) na Equação (2.5), obtém-se a distância Euclidiana na Equação (2.6).

$$D_e(p, q) = \sqrt{(s - u)^2 + (t - v)^2} \quad (2.6)$$

2.5 – Filtragem Espacial

A *filtragem espacial* é uma técnica de processamento digital de imagem que utiliza os níveis de cinza dos pixels de uma certa região da imagem para gerar pixels com um novo valor de nível de cinza em uma nova imagem.

Máscara (também chamada de *template*, *janela* ou *filtro*) é definida como uma matriz onde todos os pixels da mesma possuem valores de uma determinada região da imagem. A Figura 2.4 mostra uma máscara w de tamanho 3×3 .

$w(-1,-1)$	$w(-1,0)$	$w(-1,1)$
$w(0,-1)$	$w(0,0)$	$w(0,1)$
$w(1,-1)$	$w(1,0)$	$w(1,1)$

Figura 2.4 – Máscara w de tamanho 3×3 .

O tamanho de uma máscara w é representado por $m \times n$, onde m indica o número de linhas e n o número de colunas. Normalmente em processamento digital de imagem, a dimensão de uma máscara é ímpar, no entanto ela pode possuir dimensão par.

O valor do pixel $g(x, y)$ resultante da filtragem espacial de uma imagem $f(x, y)$ por uma máscara w de tamanho $m \times n$, é calculado pela Equação (2.7).

$$g(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) \cdot f(x + s, y + t) \quad (2.7)$$

onde:

$$a = \frac{m-1}{2}$$

$$b = \frac{n-1}{2}$$

Todos os pixels da imagem $f(x, y)$ são varridos pela máscara w e o resultado é uma nova imagem $g(x, y)$. A Figura 2.5 mostra uma máscara w posicionada sobre uma região qualquer de uma imagem. Depois de calcular o valor do pixel da nova imagem $g(x, y)$, a máscara muda para a próxima posição até que toda a imagem seja varrida.

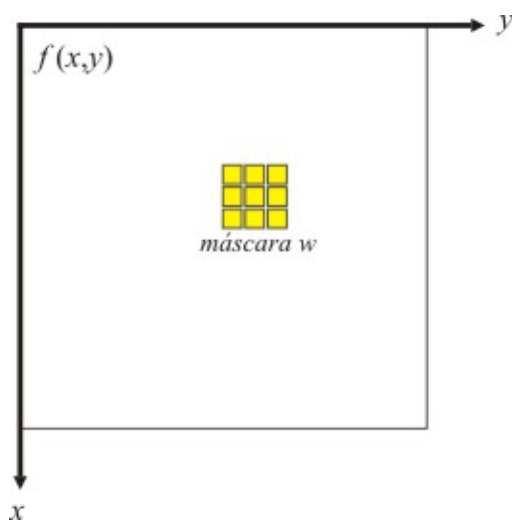


Figura 2.5 – Filtragem de $f(x, y)$ por uma máscara w .

2.6 – Convolução Discreta

A *convolução discreta* de uma imagem $f(x, y)$ por uma função $g(x, y)$ é obtida pela Equação (2.8).

$$f(x, y) * g(x, y) \triangleq \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) \cdot g(x - m, y - n) \quad (2.8)$$

Quando duas funções $f(x, y)$ e $g(x, y)$ de tamanho $M_1 \times N_1$ e $M_2 \times N_2$, respectivamente são convoluídas, o resultado é uma nova função discreta de tamanho $(M_1 + M_2 - 1) \times (N_1 + N_2 - 1)$.

Uma outra notação proposta por Jain [16], definida na Equação (2.9), também pode ser utilizada no cálculo da convolução discreta de duas funções.

$$g(x, y; m, n) \triangleq g(x - m, y - n) \quad (2.9)$$

Utilizando a Equação (2.9), a Equação (2.8) pode ser escrita pela Equação (2.10):

$$f(x, y) * g(x, y) \triangleq \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) \cdot g(x, y; m, n) \quad (2.10)$$

A expressão $g(x - m, y - n)$ na Equação (2.9) estabelece duas características da função $g(x, y)$: primeiro $g(x - m, y - n)$ translada todos os elementos de $g(x, y)$ de $-m$ e $-n$ em relação a origem; segundo $g(x - m, y - n)$ ocasiona o rebatimento dos valores de $g(x, y)$ em relação aos eixos x e y . A Figura 2.6 mostra o resultado da expressão $g(x - m, y - n)$ quando, por exemplo, x e y são iguais a zero.

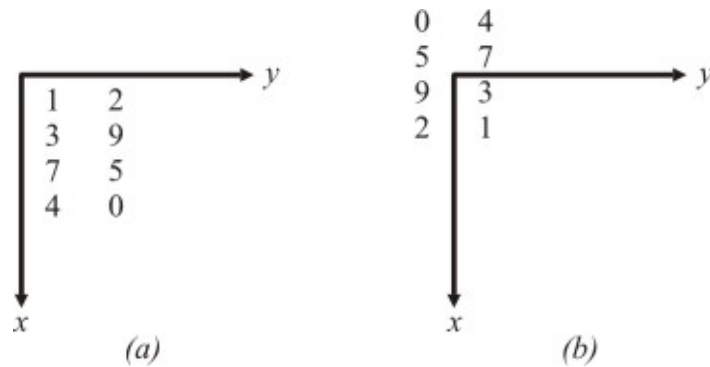


Figura 2.6 – (a) Sinal original; (b) sinal rebatido e translado ($x = y = 0$).

2.7 – Cálculo do Gradiente

Para funções contínuas de duas ou mais variáveis, o *gradiente* é um vetor em que sua direção é a maior taxa de variação de uma função.

Considerando uma função bidimensional $f(x, y)$, o cálculo das componentes x e y do gradiente é obtido a partir das derivadas parciais de $f(x, y)$ em relação a x e a y , respectivamente, como mostrado na Equação (2.11).

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (2.11)$$

O módulo e a direção do gradiente são encontrados utilizando-se as Equações (2.12) e (2.13), respectivamente.

$$|\nabla f| = \sqrt{G_x^2 + G_y^2} \quad (2.12)$$

$$\alpha(x, y) = \tan^{-1}\left(\frac{G_y}{G_x}\right) \quad (2.13)$$

A definição de gradiente não se restringe somente a funções contínuas, ela pode ser aplicada a funções discretas, como por exemplo, a uma imagem $f(x, y)$. Da mesma forma, o gradiente para funções discretas indica a direção da maior variação da função $f(x, y)$.

Uma aproximação para o cálculo do gradiente para funções discretas é obtida pela Equação (2.14).

$$\nabla f = |G_x| + |G_y| \quad (2.14)$$

Outra característica interessante do gradiente é que a direção perpendicular à direção dele possui uma variação nula na função $f(x, y)$.

Para o cálculo do gradiente de uma imagem $f(x, y)$ pode-se utilizar a técnica de *filtragem espacial*. Para o cálculo das componentes do gradiente de uma imagem existem algumas máscaras, chamadas de *operadores*. A Figura 2.7 mostra alguns operadores citados por Jain [16] utilizados em uma imagem.

Roberts	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ (a)	$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$ (b)
Prewitt	$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$ (c)	$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$ (d)
Sobel	$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$ (e)	$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$ (f)
Isotrópico	$\begin{bmatrix} -1 & -\sqrt{2} & -1 \\ 0 & 0 & 0 \\ 1 & \sqrt{2} & 1 \end{bmatrix}$ (g)	$\begin{bmatrix} -1 & 0 & 1 \\ -\sqrt{2} & 0 & \sqrt{2} \\ -1 & 0 & 1 \end{bmatrix}$ (h)

Figura 2.7 – Operadores utilizados para o cálculo do gradiente em imagens.

Como os operadores para o cálculo do gradiente são baseados em derivadas parciais e a diferenciação aumenta o ruído, os operadores de Sobel fornecem resultados melhores do que os outros operadores. A vantagem dos operadores de Sobel em relação aos outros operadores, é que além de obter as componentes do gradiente, eles também suavizam a imagem. A Figura 2.8 mostra o resultado do cálculo do gradiente de uma imagem utilizando os operadores de Sobel.



Figura 2.8 – Aplicação dos operadores de Sobel. (a) imagem original; (b) componente G_x ; (c) componente G_y ; (d) gradiente da imagem original.

A Figura 2.8(b) mostra o resultado da componente de G_x e como esperado as linhas horizontais da imagem resultante foram mais realçadas. Na Figura 2.8(c) as linhas verticais foram enfatizadas, pois trata-se da componente G_y . A Figura 2.8(d) ilustra o resultado das duas componentes G_x e G_y , simultaneamente.

2.8 – Cálculo da Orientação de Campo

Orientação de campo é a direção de cada pixel (x, y) da imagem. Nesta seção, essa orientação é representada pelo segmento de reta.

Considere um conjunto de segmentos de retas, como mostrado na Figura 2.9. Existe um problema a ser resolvido: como determinar a orientação dominante desse conjunto de segmentos ?

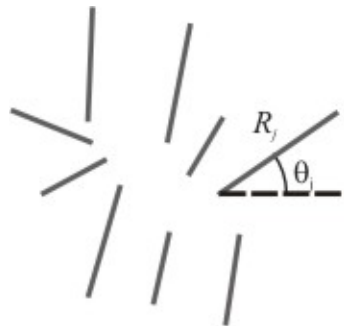


Figura 2.9 – Segmentos de retas em uma região de uma imagem

A primeira proposta de Rao [17] para determinar a orientação dominante de um conjunto de segmentos de retas foi realizar a soma vetorial desses segmentos e encontrar a direção resultante. Contudo, essa proposta não proporcionou bons resultados por duas razões. A primeira é que os segmentos de retas não possuem uma única direção, como mostrado na Figura 2.10. Se o ponto A é adotado como início do vetor, a direção é θ conforme ilustra a Figura 2.10(b). Se em vez de A , é adotado o ponto B como início do vetor, a direção é $\theta + \pi$ como mostrado na Figura 2.10(c).

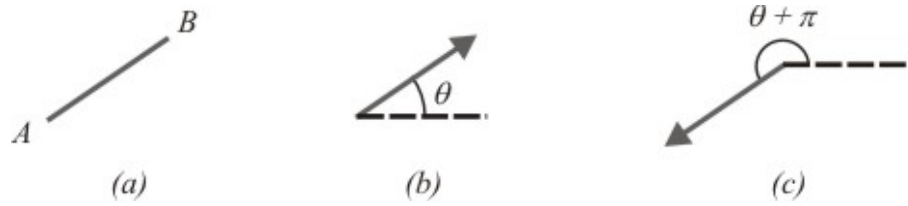


Figura 2.10 – (a) Segmento de reta AB; (b) direção θ considerando A como início do vetor;
 (c) direção $\theta + \pi$ considerando B como início do vetor.

A segunda razão da proposta de Rao [17] não ter proporcionado bons resultados é se a escolha dos pixels que as direções são medidas é oposta, ocorre um cancelamento desses segmentos de retas em vez deles contribuírem na orientação dominante.

2.9 – Determinação da Orientação de Campo Utilizando o Gradiente

Para eliminar o problema encontrado por Rao [17] no cálculo da orientação de campo de uma imagem $f(x, y)$, ele utiliza um outro método. Este método é descrito a seguir.

Considere, um vetor no plano complexo formado pela combinação das componentes x e y do gradiente da imagem $f(x, y)$, como mostrado na Equação (2.15) ou representado de forma polar pela Equação (2.16).

$$G = G_x + iG_y \quad (2.15)$$

$$G = Re^{i\theta} \quad (2.16)$$

onde:

$$R = \sqrt{G_x^2 + G_y^2}$$

$$\theta = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$

A Figura 2.11 mostra o vetor e suas componentes x e y representadas no plano xy .

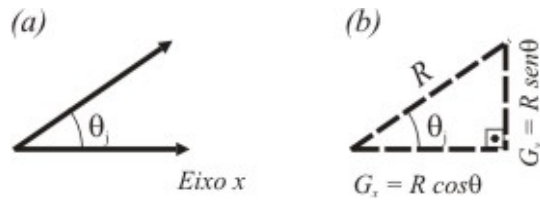


Figura 2.11 – (a) Representação do vetor no plano xy ; (b) componentes G_x e G_y representadas nesse plano.

As componentes G_x e G_y podem ser calculadas utilizando-se as Equações (2.17) e (2.18).

$$G_x = R \cos \theta \quad (2.17)$$

$$G_y = R \sen \theta \quad (2.18)$$

O quadrado do vetor G é $R^2 e^{2i\theta}$. Portanto, os vetores em direções opostas reforçam uns aos outros.

Considere $J(x,y)$ o vetor gradiente quadrático na posição (x, y) calculado na forma cartesiana e polar utilizando-se as Equações (2.19) e (2.20), respectivamente.

$$J(x, y) = [G(x, y)]^2 \quad (2.19)$$

$$J(R, \theta) = (R \cdot e^{i\theta})^2 = R^2 \cdot e^{2i\theta} \quad (2.20)$$

As componentes x e y do vetor gradiente quadrático são indicadas por J_x e J_y , respectivamente.

A parte real J_x do vetor gradiente quadrático é calculada a partir da parte real de J , então:

$$J_x = R^2 \cdot \cos 2\theta = R^2 (\cos^2 \theta - \sin^2 \theta) \quad (2.21)$$

$$J_x = (R \cos \theta)^2 - (R \sin \theta)^2 \quad (2.22)$$

Substituindo-se as Equações (2.17) e (2.18) na Equação (2.22), o resultado é a Equação (2.23).

$$J_x = G_x^2 - G_y^2 \quad (2.23)$$

O cálculo da parte imaginária J_y é calculado pela parte imaginária de J , então:

$$J_y = R^2 \cdot \sin 2\theta = R^2 (2 \cos \theta \cdot \sin \theta) \quad (2.24)$$

$$J_y = 2 \cdot (R \cos \theta) \cdot (R \sin \theta) \quad (2.25)$$

Substituindo-se as Equações (2.17) e (2.18) na Equação (2.25), o valor da componente J_y é a Equação (2.26).

$$J_y = 2 \cdot G_x \cdot G_y \quad (2.26)$$

Para obter uma orientação de campo mais suavizada (mais próxima da orientação dominante) para uma imagem utilizando-se os pixels pertencentes a região em torno do pixel (x, y) , basta considerar um bloco de tamanho $W \times W$ e o centro desse bloco localizado no ponto (x, y) . Logo, as somas vetoriais das componentes $J_x(V_x(x, y))$ e $J_y(V_y(x, y))$ nessa vizinhança, são obtidas pelas Equações (2.27) e (2.28), respectivamente.

$$V_x(x, y) = \sum_{u=x-\frac{W}{2}}^{x+\frac{W}{2}} \sum_{v=y-\frac{W}{2}}^{y+\frac{W}{2}} \left[(G_x(u, v))^2 - (G_y(u, v))^2 \right] \quad (2.27)$$

$$V_y(x, y) = \sum_{u=x-\frac{W}{2}}^{x+\frac{W}{2}} \sum_{v=y-\frac{W}{2}}^{y+\frac{W}{2}} \left[2G_x(u, v)G_y(u, v) \right] \quad (2.28)$$

Depois de calcular os valores de $V_x(x, y)$ e $V_y(x, y)$, pode-se definir a orientação de campo do pixel (x, y) pela Equação (2.29).

$$\theta(x, y) = \frac{1}{2} \tan^{-1} \left(\frac{V_y(x, y)}{V_x(x, y)} \right) \quad (2.29)$$

onde:

$\theta(x, y)$ – orientação de campo para a posição (x, y) .

Na Equação (2.29) é necessário a divisão por dois porque $\tan^{-1} \left(\frac{V_y(x, y)}{V_x(x, y)} \right)$ é a orientação do vetor gradiente quadrático, ou seja, a direção de J . A Equação (2.20) mostra que a direção de J é duas vezes a de G , então para obter a direção de G a partir de J basta dividi-la por dois.

2.10 – Considerações Finais deste Capítulo

Este capítulo apresentou alguns conceitos fundamentais em processamento digital de imagem para a extração das características necessárias a serem utilizadas no reconhecimento de impressões digitais.

Este capítulo descreveu um modelo simples de uma imagem, vizinhança de um pixel, distância Euclidiana, filtragem espacial, convolução discreta, cálculo do gradiente e determinação da orientação de campo de uma imagem digital.

O próximo capítulo descreverá a impressão digital e suas características, o cálculo da orientação de campo para cada pixel da imagem de impressão digital, o cálculo do nível de

consistências dessa orientação e a determinação da área de interesse dessa imagem. São descritas as seguintes operações em relação as saliências: extração, afinamento e remoção das falhas. Depois são apresentadas a detecção de minúcias, a correção de falhas presentes em saliências de uma impressão digital, a determinação das coordenadas dos pixels de uma saliência, a remoção e a conversão das saliências em sinais discretos unidimensionais e o cálculo da distância média entre as saliências.

CAPÍTULO III

MÉTODOS DE EXTRAÇÃO DE CARACTERÍSTICAS EM

IMAGENS DE IMPRESSÕES DIGITAIS

3.1 – Introdução

O objetivo deste capítulo é mostrar as técnicas de extração de características de imagens de impressão digital. Essas características são utilizadas no reconhecimento de impressões digitais.

Este capítulo mostra a impressão digital e suas características, o cálculo da orientação de campo para cada pixel da imagem de impressão digital, o cálculo do nível de consistência dessa orientação e a determinação da área de interesse dessa imagem. São descritas as seguintes operações em relação as saliências: extração, afinamento e remoção das falhas. Depois são apresentadas a detecção de minúcias, a correção de falhas presentes em saliências de uma impressão digital, a determinação das coordenadas dos pixels de uma saliência, a remoção e a conversão das saliências em sinais discretos unidimensionais e o cálculo da distância média entre as saliências.

3.2 – A Impressão Digital e suas Características

As impressões digitais são as linhas presentes nos dedos de cada ser humano e essa é única para cada um, ou seja, não existem duas pessoas que possuem a mesma forma com que as linhas da impressão se dispõem em seus dedos. Por esse motivo é que a impressão digital vem sendo utilizada para a identificação de pessoas para várias finalidades, como por exemplo, o controle de ponto dos funcionários de uma empresa, a identificação de um criminoso e o acesso de pessoas a áreas restritas de extrema segurança.

A impressão digital é formada por *vales* e *saliências*. A Figura 3.1 mostra um exemplo de uma região de uma impressão digital de uma pessoa onde está realçada na imagem uma *saliência* e um *vale*. Nessa figura os vales são representados pelas linhas mais claras enquanto as *saliências* pelas linhas mais escuras.

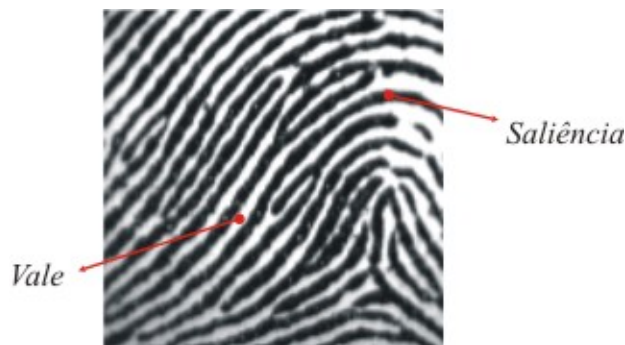


Figura 3.1 – Região de uma impressão digital.

A impressão digital pode ser classificada em função da forma com que suas *saliências* e seus *vales* estão dispostos na superfície do dedo. Nessa classificação existem seis tipos que são mostrados na Figura 3.2.

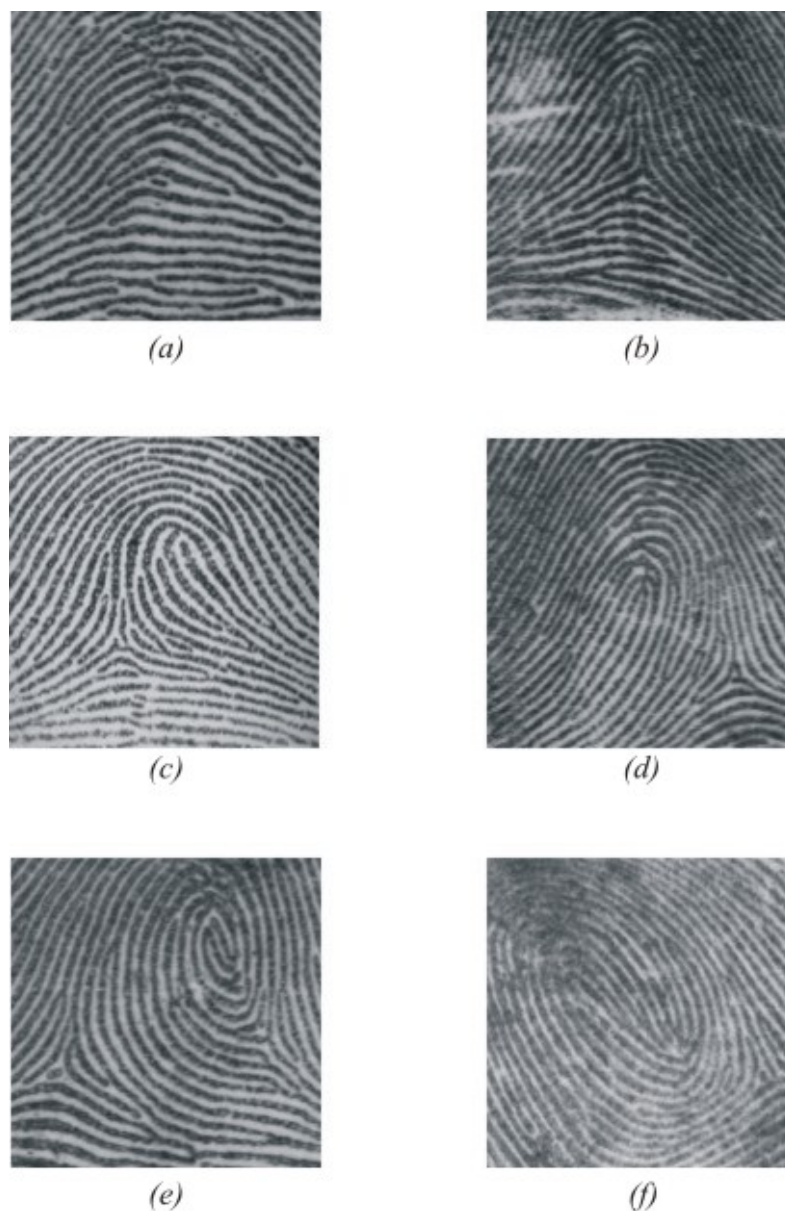


Figura 3.2 – Tipos de impressão digital. (a) arco; (b) arco “tented”; (c) loop à esquerda; (d) loop à direita; (e) “whorl”; (f) loop gêmeos.

Para o reconhecimento de impressões digitais são utilizadas as *minúcias*, que são características das *saliências*. Existem dois tipos de minúcias: as *terminações de saliências* e as *bifurcações de saliências*. A Figura 3.3 mostra uma terminação de saliência e uma bifurcação de saliência onde cada uma está realçada por um círculo amarelo.

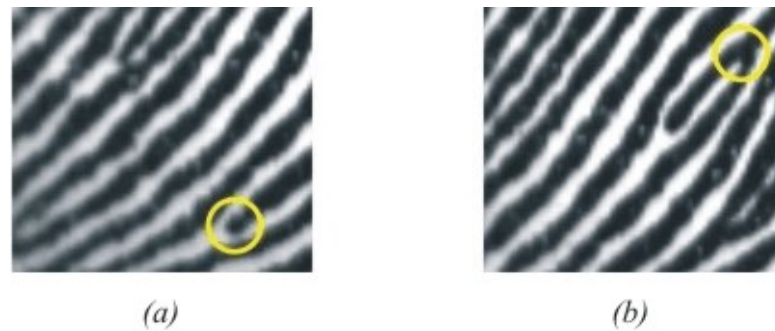


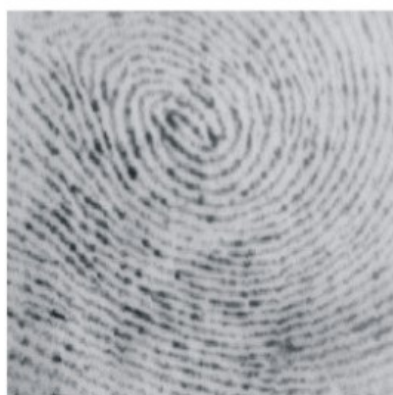
Figura 3.3 – Tipos de minúcias. (a) terminação de saliência; (b) bifurcação de saliência.

Existem alguns problemas que ocorrem em um algoritmo de reconhecimento de impressões digitais que dificultam a extração das informações. Os mais graves são:

- *Contato não-uniforme* – no caso em que as impressões digitais são coletadas por *scanners*, um contato não uniforme com a área de captura, gera na região de menor pressão, uma imagem mais clara que quase não são evidenciadas as saliências dessa região, como mostrado na Figura 3.4(b);
- *Superfície da impressão danificada* – este problema é devido a trabalhos manuais, doenças de pele, acidentes e outras situações que possam danificar a superfície da impressão mudando a estrutura das saliências e criando assim falsas minúcias; e
- *Translação e rotação* – um indivíduo pode posicionar o dedo sobre o scanner da impressão em posições diferentes a que está registrada no banco de dados, ou seja, translado e rotacionado, como mostrado na Figura 3.5.



(a)



(b)

Figura 3.4 – Imagens de uma mesma impressão obtidas por um scanner. (a) imagem com uma pressão maior e mais distribuída do que a da letra (b).



(a)



(b)

Figura 3.5 – Imagens da mesma impressão, a letra (b) está translada e rotacionada em relação a letra (a).

3.3 – Cálculo da Orientação de Campo de uma Imagem de Impressão Digital

Uma das características necessárias para o reconhecimento de impressões digitais é a *orientação de campo* ou também chamada de *campo vetorial*. Os passos para o cálculo da orientação de campo desenvolvido por Rao [17] são:

1º – Dividi-se a imagem da impressão digital em blocos de tamanho $W \times W$;

2º – Calcula-se as componentes x e y do gradiente para cada pixel de cada bloco da imagem;

3º – Estima-se a orientação de campo de cada pixel (x,y) , utilizado-se as Equações (3.1), (3.2) e (3.3).

$$V_x(x, y) = \sum_{u=x-\frac{W}{2}}^{x+\frac{W}{2}} \sum_{v=y-\frac{W}{2}}^{y+\frac{W}{2}} \left[(G_x(u, v))^2 - (G_y(u, v))^2 \right] \quad (3.1)$$

$$V_y(x, y) = \sum_{u=x-\frac{W}{2}}^{x+\frac{W}{2}} \sum_{v=y-\frac{W}{2}}^{y+\frac{W}{2}} \left[2G_x(u, v)G_y(u, v) \right] \quad (3.2)$$

$$\theta(x, y) = \frac{1}{2} \tan^{-1} \left(\frac{V_y(x, y)}{V_x(x, y)} \right). \quad (3.3)$$

onde:

W – tamanho do bloco.

G_x e G_y – magnitudes das componentes do gradiente na direção x e y , respectivamente.

$\theta(x, y)$ – orientação de campo para o pixel de coordenada (x, y) .

O terceiro passo do algoritmo de cálculo de orientação de campo desenvolvido por Rao [17] indica que é necessário varrer toda a imagem e calcular V_x e V_y para cada pixel, obtendo-se assim o valor de θ para a posição (x, y) .

A Figura 3.6 ilustra uma imagem de tamanho 18×18 pixels onde os quadrados brancos são os pixels da imagem e os quadrados vermelhos são os blocos que a imagem é dividida, como estabelecido no 1º passo do algoritmo de cálculo da orientação de campo de Rao [17]. Nessa figura cada bloco possui um $W = 9$.

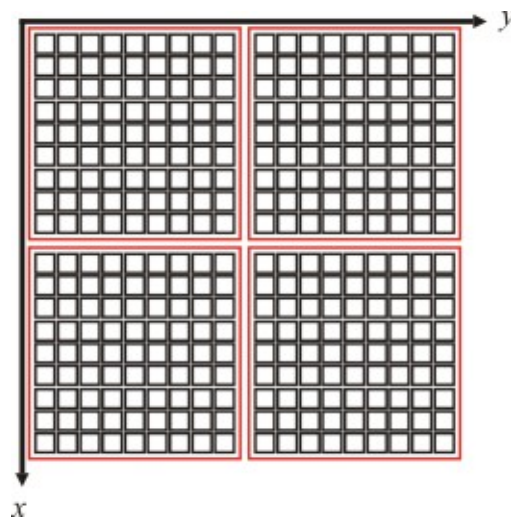


Figura 3.6 – Imagem dividida em quatro blocos.

A divisão da imagem em blocos tem a vantagem de evitar que um ruído presente em um determinado bloco não seja transmitido aos blocos vizinhos no momento do cálculo do gradiente, pois esse cálculo utiliza derivadas e a diferenciação tende a aumentar o ruído. Cada bloco é tratado como uma imagem independente, então os pixels do bloco que está sujeito ao ruído não são utilizados pelos outros blocos para o cálculo do gradiente, assim o ruído não se propaga. A Figura 3.7 mostra a questão da não-propagação do ruído em outros blocos da imagem.

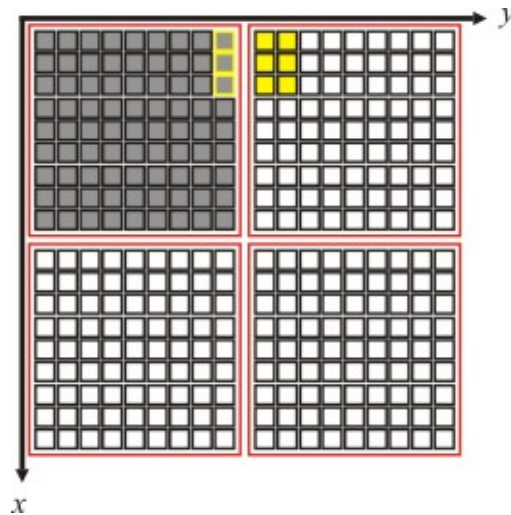


Figura 3.7 – A não propagação do ruído em outros blocos da imagem.

Na Figura 3.7, o bloco com a presença de ruído é representado pelos pixels que estão realçados pela cor cinza. Suponha que para o cálculo do gradiente é utilizado a técnica de filtragem espacial onde os operadores possuem tamanho igual a 3×3 . Se a imagem não fosse dividida em blocos, para cada pixel da primeira coluna do bloco superior-direito seria utilizado três pixels da última coluna do bloco superior-esquerdo. Essa figura mostra, por exemplo, que para o cálculo do valor dos pixels da segunda linha e da primeira coluna,

localizado no bloco superior-direito é necessário utilizar três pixels da última coluna (realçados pelas bordas amarelas) no bloco superior-esquerdo.

Considerando que existe ruído somente no bloco superior-esquerdo, a Figura 3.8 mostra na cor vermelha, todos pixels que são afetados pelo ruído devido à vizinhança com o bloco ruidoso, considerando que os operadores para o cálculo do gradiente possuem dimensões iguais a 3×3 e a imagem não for dividida em blocos.

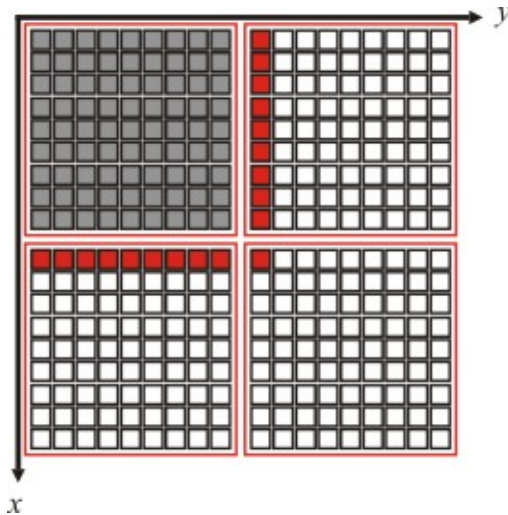


Figura 3.8 – Pixels afetados pelo ruído presentes no bloco superior-esquerdo.

3.3.1 – Cálculo do somatório unidimensional utilizando a propriedade da comutação

O cálculo da orientação de campo foi mostrado na Seção 3.3 deste capítulo. Neste cálculo para encontrar os ângulos para cada pixel da imagem é necessário realizar duas somatórias, representadas pelas Equações (3.1) e (3.2). Neste trabalho foi desenvolvido um algoritmo para diminuir o número de adições nas somatórias dessas equações utilizando a propriedade da *comutação*.

Suponha que **A**, **B**, e **C** sejam números reais. A propriedade da *comutação* pode ser exemplificada pela Equação (3.4).

$$(\mathbf{A} + \mathbf{B}) + \mathbf{C} = (\mathbf{A} + \mathbf{C}) + \mathbf{B} = (\mathbf{B} + \mathbf{C}) + \mathbf{A} \quad (3.4)$$

Para ilustrar o uso da propriedade da comutação na redução do número de adições em uma somatória, considere um sinal discreto unidimensional $f(x)$, como mostrado na Figura 3.9. Nesta figura cada quadrado com seu respectivo número dentro representa uma amostra desse sinal na posição x .

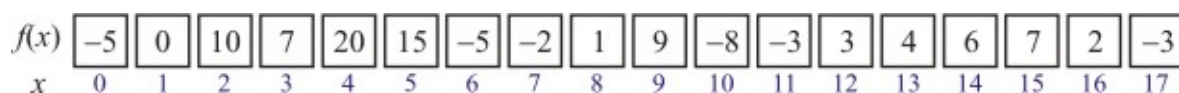


Figura 3.9 – Sinal discreto $f(x)$.

Suponha que se deseja calcular a expressão da Equação (3.5).

$$g(x) = \sum_{i=x-a}^{x+a} f(i) \quad (3.5)$$

onde:

$f(i)$ – sinal discreto mostrado na Figura 3.9.

$g(x)$ – sinal discreto resultante da somatória representada pela Equação (3.5).

a – constante inteira, que indica que existem a elementos à esquerda e a elementos à direita do elemento localizado na posição x .

Como $f(i)$ possui um comprimento igual a L , que no exemplo da Figura 3.9 é 18, então $g(x)$ também possui um comprimento igual a L . Quando o índice i está fora do intervalo $[0, L - 1]$, o valor de $f(i)$ é considerado igual a zero.

O resultado da Equação (3.5), para todos os valores de x variando de 0 a $L - 1$, é mostrado na Figura 3.10.

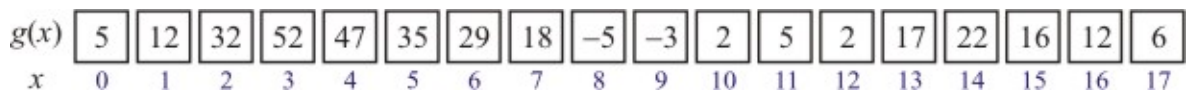


Figura 3.10 – Sinal discreto $g(x)$ de comprimento $L = 18$ e $a = 2$.

A finalidade básica da utilização da propriedade da comutação no cálculo da orientação de campo é o reaproveitamento das somatórias já calculadas. Quando é calculado o valor de $g(x)$ para uma determinada posição x , a somatória da posição seguinte $x + 1$ é quase idêntica a somatória anterior, exceto por dois valores.

Para exemplificar a utilização da propriedade da comutação, considere que se deseja calcular o valor de $g(7)$. Logo a Equação (3.5) é igual a Equação (3.6) para $a = 2$.

$$g(7) = \sum_{i=5}^9 f(i) = f(5) + f(6) + f(7) + f(8) + f(9) = 18 \quad (3.6)$$

Agora é necessário calcular o valor de $g(8)$ que resulta na Equação (3.7) para $a = 2$.

$$g(8) = \sum_{i=6}^{10} f(i) = f(6) + f(7) + f(8) + f(9) + f(10) = -5 \quad (3.7)$$

As Figuras 3.11(a) e 3.11(b) mostram os elementos utilizados para calcular o valor de $g(7)$ e os elementos necessários para o cálculo de $g(8)$, respectivamente. Como pode-se notar nessas figuras, existem alguns elementos em comum que são: $f(6)$, $f(7)$, $f(8)$ e $f(9)$. A diferença entre as somatórias que resulta em $g(7)$ e $g(8)$ é que o primeiro elemento da somatória de $g(7)$ não faz parte da somatória de $g(8)$ e que o último elemento da somatória de $g(8)$ não faz parte da somatória de $g(7)$. A Figura 3.11(c) mostra os elementos comuns as duas somatórias de $g(7)$ e $g(8)$ na cor amarela e os elementos diferentes na cor cinza. Pode-se notar que existe um desperdício computacional a respeito das somatórias, pois para o próximo elemento calculado, a somatória parte do zero em vez de aproveitar a somatória anterior e reduzir o número de operações aritméticas, isto ocorre no algoritmo de orientação de campo desenvolvido por Jain e outros [1], ou seja, esse algoritmo não utiliza a propriedade da comutação. Porém o algoritmo desenvolvido neste trabalho utiliza essa propriedade para reduzir o tempo de processamento.

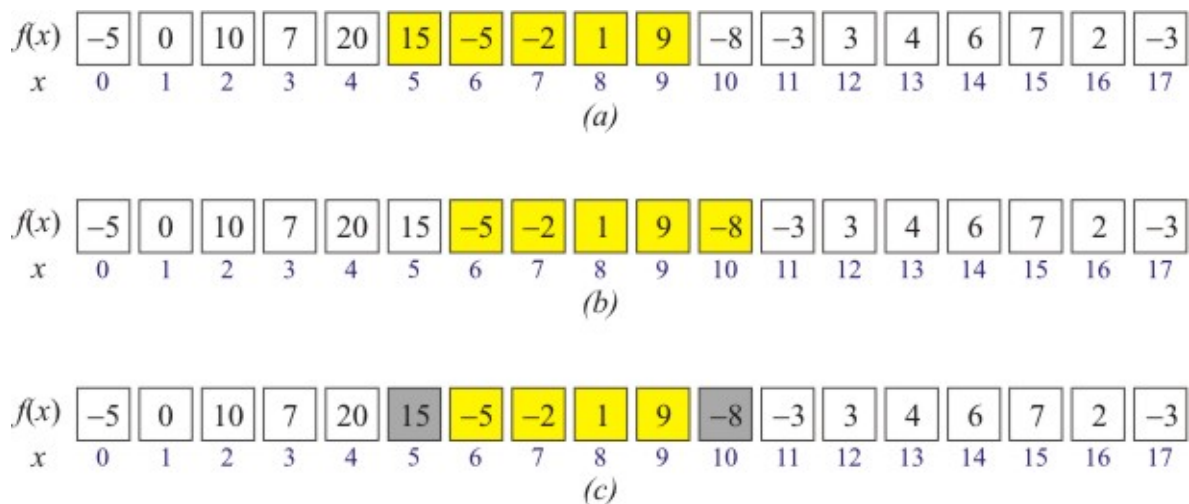


Figura 3.11 – Elementos pertencentes. (a) $g(7)$; (b) $g(8)$; (c) $g(7)$ e $g(8)$.

Então, utilizando-se a propriedade da comutação para toda a posição $x + 1$ pode-se reaproveitar o cálculo da somatória da posição x , onde é necessário retirar o primeiro elemento da somatória de x e adicionar o último elemento referente a somatória de $x + 1$. Por exemplo, considerando-se novamente o caso mostrado na Figura 3.11, o valor de $g(8)$ pode ser encontrado utilizando-se a Equação (3.8).

$$g(8) = f(10) - f(5) + \sum_{i=5}^9 f(i) \quad (3.8)$$

A Equação (3.8) também pode ser escrita pela Equação (3.9).

$$g(8) = f(10) - f(5) + g(7) \quad (3.9)$$

De forma geral, o cálculo do sinal discreto $g(x)$ pode ser representado pela Equação (3.10).

$$g(x) = \begin{cases} \sum_{i=0}^a f(i), & \text{se } x = 0 \\ g(x-1) + f(x+a) - f(x-a-1), & \text{se } 1 \leq x \leq L-1 \end{cases} \quad (3.10)$$

A Equação (3.10) mostra que é necessário calcular a somatória uma única vez (quando x é igual a zero). A somatória para $x = 0$ não precisa ter o valor inicial de $i = x - a$, pois resulta em valores de i negativo, que possuem $f(i)$ igual a zero. O elemento $g(x-1)$ representa a somatória já calculada, $f(x+a)$ é o novo elemento adicionado a somatória e $f(x-a-1)$ é o

primeiro elemento que compõe a somatória anterior. O que garante que se pode somar valores as somatórias anteriores é justamente a propriedade da comutação.

Para verificar a eficiência da utilização da propriedade da comutação no cálculo da orientação de campo de uma imagem de impressão digital, segue a dedução das equações do número de somas realizadas pelas somatórias das Equações (3.5) e (3.10). Para a dedução dessas equações, considere a somatória da Equação (3.5) e um sinal de comprimento L .

O valor de N_{SC} representa o número de somas calculadas por uma somatória sem utilizar a propriedade da comutação, esse valor pode ser encontrado na Equação (3.11).

$$N_{SC} = n^{\circ} \text{ de adições por posição} \times n^{\circ} \text{ de posições} \quad (3.11)$$

Na Equação (3.5), a indica que existem a elementos à esquerda e a elementos à direita do elemento central da somatória, no total existem $2a + 1$ elementos que participam da somatória de uma posição x . Como o número de adições é o total de elementos menos um, então o valor do $n^{\circ} \text{ de adições por posição}$ é igual $2a$ e o valor do $n^{\circ} \text{ de posições}$ é igual L (comprimento do sinal).

Substituindo-se os valores do $n^{\circ} \text{ de adições por posição}$ e do $n^{\circ} \text{ de posições}$, o resultado da Equação (3.11) é a Equação (3.12).

$$N_{SC} = 2aL \quad (3.12)$$

O valor N_{CC} representa o número de somas calculadas por uma somatória que utiliza a propriedade da comutação, esse valor pode ser encontrado utilizando-se a Equação (3.13).

$$N_{CC} = n^{\circ} \text{ de elementos somados na } 1^{\text{a}} \text{ posição} + n^{\circ} \text{ de somas das posições restantes} \quad (3.13)$$

Pode-se verificar na Equação (3.10) que se $x = 0$ é calculada a soma de $(a + 1)$ elementos, então o número de adições para a primeira posição é igual a a . Para as demais $(L - 1)$ posições são calculadas duas somas. O valor de N_{CC} é igual a Equação (3.14).

$$N_{CC} = a + 2(L - 1) \quad (3.14)$$

Para um sinal de comprimento $L = 256$ amostras, a Tabela 3.1 mostra o número de adições com e sem utilizar a propriedade da comutação.

Tabela 3.1 – Número de adições com e sem utilizar a propriedade da comutação.

a	N_{SC}	N_{CC}	$\frac{N_{CC}}{N_{SC}} \times 100$
2	1.024	512	50,00 %
5	2.560	515	20,11 %
8	4.096	518	12,64 %
10	5.120	520	10,15 %
16	8.192	526	6,42 %
20	10.240	530	5,17 %

Na Tabela 3.1 a razão $\frac{N_{CC}}{N_{SC}}$ pode ser definida como uma vantagem computacional, pois relaciona o número de adições de N_{CC} em relação a N_{SC} . A vantagem computacional mostra, em porcentagem que a quantidade N_{CC} tem N_{SC} como referência. Para exemplificar,

considerando nessa tabela o comprimento do sinal igual a 256, e a igual a 16, a utilização da propriedade de comutação demanda 6,42 % menos adições do que a somatória que não utiliza essa propriedade.

3.3.2 – Aplicando a propriedade da comutação no cálculo da orientação de campo

Considere um sinal bidimensional, como por exemplo, uma imagem $f(x, y)$. O algoritmo desenvolvido nesta dissertação que utiliza a propriedade da comutação para reduzir o número de operações das somatórias unidimensionais, também pode ser aplicado para somatórias bidimensionais como mostrado na Equação (3.15).

$$g(x, y) = \sum_{i=x-a}^{x+a} \sum_{j=y-b}^{y+b} f(i, j) \quad (3.15)$$

onde:

a e b – constantes inteiras.

$f(x, y)$ – sinal bidimensional de dimensão $M \times N$.

$g(x, y)$ – sinal bidimensional de dimensão $M \times N$ resultante da somatória.

É importante lembrar que a função $f(x, y)$ é definida pela Equação (3.16).

$$f(x, y) = \begin{cases} f(a, b), & \text{se } 0 \leq a \leq M-1 \text{ e } 0 \leq b \leq N-1 \\ 0, & \text{caso contrário} \end{cases} \quad (3.16)$$

As Equações (3.1) e (3.2) são utilizadas para calcular as somatórias de um sinal bidimensional. A função que calcula essas somatórias que resulta nos valores de V_x e V_y precisam receber os valores de G_x e G_y , que são as componentes do gradiente nas direções x e y , respectivamente.

Os passos do algoritmo desenvolvido neste trabalho para o cálculo da somatória bidimensional que utiliza a propriedade da comutação são:

- 1º – Define-se dois vetores V_{xs} e V_{ys} unidimensionais, cada um com comprimento igual a N , onde todos os seus elementos inicialmente são iguais a zero. É necessário satisfazer as Equações (3.17) e (3.18) para a execução desse passo.

$$V_{xs} = \begin{cases} V_{xs}(v), & \text{se } 0 \leq v \leq N-1 \\ 0, & \text{para todo o resto} \end{cases} \quad (3.17)$$

$$V_{ys} = \begin{cases} V_{ys}(v), & \text{se } 0 \leq v \leq N-1 \\ 0, & \text{para todo o resto} \end{cases} \quad (3.18)$$

- 2º – Considere as variáveis, i, j, V_x e V_y , com valores iguais à zero;

- 3º – Calcule as somatórias das Equações (3.19) e (3.20) para j variando de zero a $N-1$;

$$V_{xs}(j) = \sum_{u=0}^{\frac{W}{2}} \left[(G_x(u, j))^2 - (G_y(u, j))^2 \right] \quad (3.19)$$

$$V_{ys}(j) = \sum_{u=0}^{\frac{W}{2}} [2G_x(u, j)G_y(u, j)] \quad (3.20)$$

4º – Considere $j = 0$;

5º – Calcule os valores de V_x e V_y utilizando as Equações (3.21) e (3.22) e vá para o 7º passo;

$$V_x = \sum_{v=0}^{y+\frac{W}{2}} V_{xs}(v) \quad (3.21)$$

$$V_y = \sum_{v=0}^{y+\frac{W}{2}} V_{ys}(v) \quad (3.22)$$

6º – Calcule os valores de V_x e V_y utilizando as Equações (3.23) e (3.24);

$$V_x = V'_x + V_{xs} \left(j + \frac{W}{2} \right) - V_{xs} \left(j - \frac{W}{2} - 1 \right) \quad (3.23)$$

$$V_y = V'_y + V_{ys} \left(j + \frac{W}{2} \right) - V_{ys} \left(j - \frac{W}{2} - 1 \right) \quad (3.24)$$

onde:

V'_x e V'_y – valores de V_x e V_y antes do 6º passo.

7º – Calcule o valor de θ para a posição (i, j) utilizando a Equação (3.25);

$$\theta(i, j) = \frac{1}{2} \tan^{-1} \left(\frac{V_y}{V_x} \right) \quad (3.25)$$

8º – Incremente j de um;

9º – Se $j < N$ vá para o 6º passo. Caso contrário, vá para o 10º passo;

10º – Some um ao valor de i e considere as variáveis V_x e V_y iguais a zero;

11º – Se $i < M$ vá para o 12º passo. Caso contrário, o cálculo está concluído, então o programa deve ser encerrado, e;

12º – Atualize os valores de $V_{xs}(j)$ e $V_{ys}(j)$ utilizando as Equações (3.26) e (3.27), para j variando de 0 a $N - 1$. Volte ao 4º passo.

$$V_{ys}(j) = V'_{ys}(j) + 2G_x \left(i + \frac{W}{2}, j \right) G_y \left(i + \frac{W}{2}, j \right) - \left[2G_x \left(i - \frac{W}{2} - 1, j \right) G_y \left(i - \frac{W}{2} - 1, j \right) \right] \quad (3.26)$$

$$V_{xs}(j) = V'_{xs}(j) + G_x \left(i + \frac{W}{2}, j \right)^2 - G_y \left(i + \frac{W}{2}, j \right)^2 - \left[G_x \left(i - \frac{W}{2} - 1, j \right)^2 - G_y \left(i - \frac{W}{2} - 1, j \right)^2 \right] \quad (3.27)$$

onde:

V'_{xs} e V'_{ys} – valores de V_{xs} e V_{ys} antes do 12º passo.

A Figura 3.12 mostra o fluxograma do algoritmo desenvolvido neste trabalho para o cálculo da orientação de campo usando a propriedade da comutação.

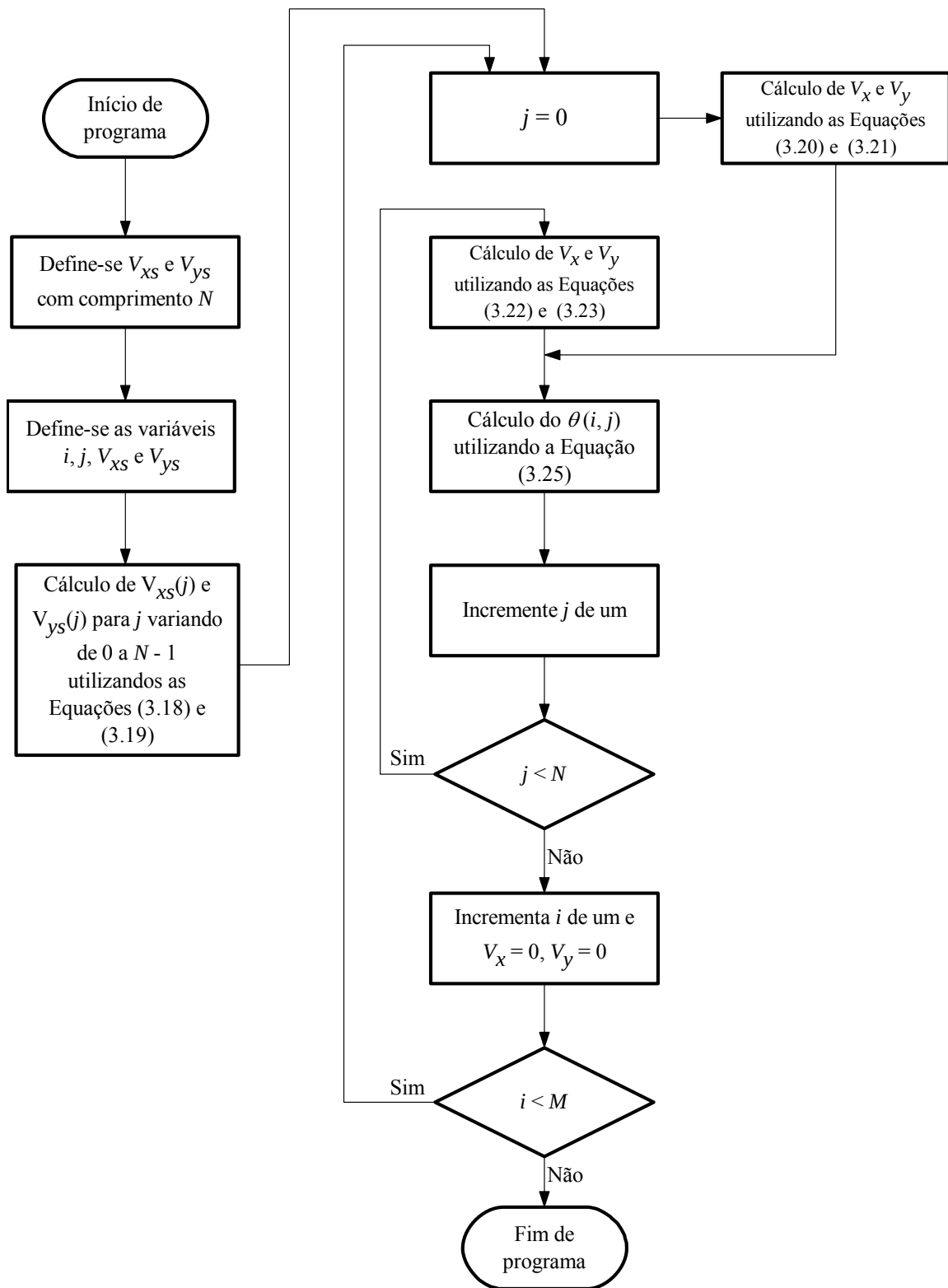


Figura 3.12 – Fluxograma do algoritmo desenvolvido neste trabalho para o cálculo da orientação de campo usando a propriedade da comutação.

O fluxograma da Figura 3.12 primeiro considera cada coluna da imagem $f(x, y)$ como sendo um sinal discreto, então define-se dois vetores unidimensionais, V_{xs} e V_{ys} , cada um de comprimento N , onde cada posição de V_{xs} e V_{ys} contém inicialmente os resultados das Equações (3.17) e (3.18). No segundo passo, são definidas as variáveis i, j, V_x e V_y onde V_x e V_y são utilizadas para armazenar temporariamente os resultados das somatórias das Equações (3.1) e (3.2) e as variáveis i e j são utilizadas para varrer toda a imagem, onde seus valores em um determinado momento, indicam a posição que está sendo processada a imagem. No terceiro passo os vetores V_{xs} e V_{ys} são inicializados, o que equivale a Equação (3.10) quando $x = 0$. O quinto passo tem a mesma função do terceiro, porém as variáveis V_x e V_y são inicializadas.

Para exemplificar o uso da propriedade da comutação em somatórias bidimensionais é realizado o cálculo da Equação (3.28).

$$g(x, y) = \sum_{i=x-\frac{W}{2}}^{x+\frac{W}{2}} \sum_{j=y-\frac{W}{2}}^{y+\frac{W}{2}} f(i, j) \quad (3.28)$$

Os valores de $f(i, j)$ utilizados na Equação (3.28) são mostrados na Figura 3.13(a).

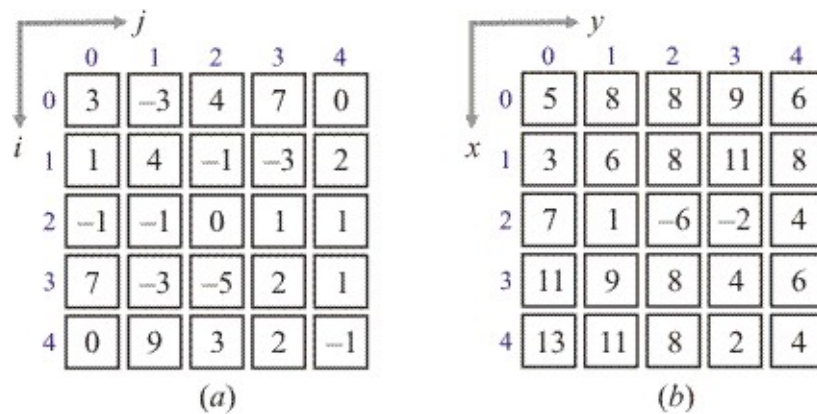


Figura 3.13 – (a) Valores de $f(i, j)$, (b) resultado da Equação (3.28).

No exemplo da Equação (3.28), os valores das variáveis x e y , variam de zero a 4. As Figuras 3.13(a) e 3.13(b) mostram os valores de $f(i, j)$ e os resultados da Equação (3.28) utilizando os valores de $f(i, j)$ para $W = 2$, respectivamente.

O exemplo do uso da propriedade da comutação em somatórias bidimensionais é mostrado somente para o cálculo de V_x para todos os valores de i e j , pois o cálculo dos valores de V_y utiliza o mesmo procedimento. Como está sendo calculado os valores de V_x , a Figura 3.13(a) representa o resultado de $[G_x(x, y)]^2 - [G_y(x, y)]^2$ para i e j variando de zero a 4.

Para o cálculo de V_x para cada valor de i e j é necessário calcular V_{xs} para a primeira linha da Figura 3.13(a) e atualizar os seus valores para as linhas restantes. A Figura 3.14, mostra os valores de V_{xs} referentes a cada linha da matriz mostrada na Figura 3.13(a).

4	1	3	4	2
---	---	---	---	---

(a)

3	0	3	5	3
---	---	---	---	---

(b)

7	0	-6	0	4
---	---	----	---	---

(c)

6	5	-2	5	1
---	---	----	---	---

(d)

7	6	-2	4	0
---	---	----	---	---

(e)

Figura 3.14 – Valores de V_{xs} para as linhas. (a) $i = 0$; (b) $i = 1$; (c) $i = 2$; (d) $i = 3$; (e) $i = 4$.

Para calcular V_x , calcula-se os valores de V_{xs} para cada valor de j (nesse exemplo da Figura 3.13(a) j varia de zero a 4), utilizando-se a Equação (3.28), onde os valores e e W são respectivamente zero e 2. Os resultados dessa equação para a primeira linha da matriz mostrada na Figura 3.13(a) são calculados, como mostram as Equações (3.29) a (3.33).

$$V_{xs}(0) = f(-1, 0) + f(0, 0) + f(1, 0) = 4 \quad (3.29)$$

$$V_{xs}(1) = f(-1, 1) + f(0, 1) + f(1, 1) = 1 \quad (3.30)$$

$$V_{xs}(2) = f(-1, 2) + f(0, 2) + f(1, 2) = 3 \quad (3.31)$$

$$V_{xs}(3) = f(-1, 3) + f(0, 3) + f(1, 3) = 5 \quad (3.32)$$

$$V_{xs}(4) = f(-1, 4) + f(0, 4) + f(1, 4) = 3 \quad (3.33)$$

Depois de calcular os valores de V_{xs} para cada coluna, calcula-se o valor de V_x para cada linha utilizando-se a Equação (3.21). Os resultados são mostrados nas Equações (3.34) a (3.38).

$$V_x(0, 0) = V_{xs}(0) + V_{xs}(1) = 5 \quad (3.34)$$

$$V_x(0, 1) = V_x(0, 0) + V_{xs}(2) - V_{xs}(-1) = 8 \quad (3.35)$$

$$V_x(0, 2) = V_x(0, 1) + V_{xs}(3) - V_{xs}(0) = 8 \quad (3.36)$$

$$V_x(0, 3) = V_x(0, 2) + V_{xs}(4) - V_{xs}(1) = 9 \quad (3.37)$$

$$V_x(0, 4) = V_x(0, 3) + V_{xs}(5) - V_{xs}(2) = 6 \quad (3.38)$$

Depois de calcular todos os valores de V_x para a primeira linha os valores de V_{xs} são atualizados, utilizando-se a Equação (3.27). Os novos valores de V_{xs} do exemplo Figura 3.13(a) são mostrados pelas Equações (3.39) a (3.43).

$$V_{xs}(0) = V'_{xs}(0) + f(2, 0) - f(-1, 0) = 3 \quad (3.39)$$

$$V_{xs}(1) = V'_{xs}(1) + f(2, 1) - f(-1, 1) = 0 \quad (3.40)$$

$$V_{xs}(2) = V'_{xs}(2) + f(2, 2) - f(-1, 2) = 3 \quad (3.41)$$

$$V_{xs}(3) = V'_{xs}(3) + f(2, 3) - f(-1, 3) = 5 \quad (3.42)$$

$$V_{xs}(4) = V'_{xs}(4) + f(2, 4) - f(-1, 4) = 3 \quad (3.43)$$

onde:

V'_{xs} – valores de V_{xs} no passo anterior.

Depois de atualizados os valores de V_{xs} , calcula-se os valores de V_x , como mostrado nas Equações (3.44) a (3.48).

$$V_x(1, 0) = V_{xs}(0) + V_{xs}(1) = 3 \quad (3.44)$$

$$V_x(1, 1) = V_x(1, 0) + V_{xs}(2) - V_{xs}(-1) = 6 \quad (3.45)$$

$$V_x(1, 2) = V_x(1, 1) + V_{xs}(3) - V_{xs}(0) = 8 \quad (3.46)$$

$$V_x(1, 3) = V_x(1, 2) + V_{xs}(4) - V_{xs}(1) = 11 \quad (3.47)$$

$$V_x(1, 4) = V_x(1, 3) + V_{xs}(5) - V_{xs}(2) = 8 \quad (3.48)$$

Depois de calcular os valores de V_x para a segunda linha, os valores de V_{xs} são atualizados e o processo se repete até que todos os valores de V_x sejam determinados.

A Figura 3.13(b) mostra os valores de V_{xs} para cada linha da matriz da Figura 3.13 (b).

O número de adições calculadas pela somatória em duas dimensões, que não utiliza a propriedade da comutação (N_{2DSC}) conforme mostrado na Equação (3.1) ou (3.2), pode ser deduzido da mesma forma que N_{SC} para uma dimensão, ou seja, pela Equação (3.49).

$$N_{2DSC} = n^o \text{ de adições por posição} \times n^o \text{ de posições} \quad (3.49)$$

Utilizando-se o mesmo raciocínio para a dedução da Equação (3.12), para cada posição (i, j) , o número de elementos utilizados na somatória é igual a $(W + 1)^2$, logo o *nº de adições por posição* é $(W + 1)^2 - 1$ que pode ser simplificado para $W^2 + 2W$. Para uma imagem de M linhas e N colunas, o número de posições é igual a MN . O resultado da Equação (3.49) é mostrado na Equação (3.50).

$$N_{2DSC} = (W^2 + 2W) M N \quad (3.50)$$

O número de adições em uma somatória em duas dimensões que utiliza a propriedade da comutação é obtido pela Equação (3.51).

$$N_{2DCC} = S_1 + S_2 + S_3 + S_4 \quad (3.51)$$

onde:

S_1 – número de adições para calcular V_{xs} para todas as colunas da imagem.

S_2 – número de adições para atualizar os valores de V_{xs} para as $(M - 1)$ linhas restantes da imagem.

S_3 – número total de adições para todas as linhas da imagem.

S_4 – número total de somas realizadas no 6º passo do fluxograma da Figura 3.12.

Para deduzir a expressão do número de adições calculadas pela somatória de duas dimensões que utiliza a propriedade da comutação (N_{2DCC}), primeiro considere somente as somatórias que resultam no valor de V_x , pois para V_y o cálculo é realizado da mesma forma.

A soma S_1 equivale ao número de adições para calcular V_{xs} para todas as colunas, como descrito no 3º passo do fluxograma da Figura 3.12. Como são $\frac{W}{2}$ somas para cada uma das N colunas, então $S_1 = \frac{WN}{2}$.

A soma S_2 refere-se às adições para atualizar os valores de V_{xs} para as $(M - 1)$ linhas restantes da imagem, como descrito no 12º passo do fluxograma da Figura 3.12. Como são duas somas por linha e são N colunas, $S_2 = 2N(M - 1)$ para obter pela primeira vez o valor de V_x .

No quinto passo do fluxograma da Figura 3.12, é necessário realizar $\frac{W}{2}$ somas para cada primeiro elemento de cada linha. Então S_3 é o número total de adições para todas as linhas da imagem. Logo $S_3 = \frac{WM}{2}$.

S_4 é o número total de somas realizadas no 6º passo. Como são realizadas duas somas por colunas (menos para a primeira coluna) e são M linhas, então, $S_4 = 2M(N - 1)$. Substituindo-se os valores de S_1 , S_2 , S_3 e S_4 na Equação (3.50) e realizando-se as devidas simplificações, obtém-se a Equação (3.52).

$$N_{2DCC} = \frac{W}{2}(M + N) + 2(2MN - M - N) \quad (3.52)$$

Para ilustrar a eficiência do algoritmo para o cálculo da orientação de campo para a somatória bidimensional que utiliza a propriedade da comutação, considere no cálculo de N_{2DSC} e N_{2DCC} que o valor de W é 16 e $M = N = L$. A Tabela 3.2 mostra o número de adições calculadas pelas somatórias bidimensionais com e sem utilizar a propriedade da comutação.

Tabela 3.2 – Número de adições para as somatórias bidimensionais considerando $W = 16$.

L	N_{2DSC}	N_{2DCC}	$\frac{N_{2DCC}}{N_{2DSC}} \times 100$
32	294.912	4.480	1,52 %
64	1.114.112	17.152	1,45 %
128	4.718.592	67.072	1,42 %
256	18.874.368	265.216	1,41 %
512	75.497.472	1.054.720	1,40 %
1024	301.989.888	4.206.592	1,40 %

Como definido para a somatória unidimensional, a razão $\frac{N_{2DCC}}{N_{2DSC}}$ é a vantagem computacional para a somatória bidimensional. Por exemplo, considerando-se na Tabela 3.2 o comprimento do sinal igual a 256 e W igual a 16, a vantagem computacional de se utilizar a propriedade da comutação é de 1,41%, ou seja, quando é utilizada essa propriedade são calculadas menos 98,59% adições de quando ela não é utilizada.

3.4 – Cálculo do Nível de Consistência da Orientação de Campo

Nível de consistência ($C(x, y)$) é o valor referente à orientação de campo de cada pixel da imagem que indica se a orientação de campo de um determinado pixel é satisfatória. Os passos do algoritmo proposto por Jain e outros [1] para calcular esse nível são:

- 1º – Calcula-se o valor de $C(x, y)$ para cada pixel da imagem, e;

$$C(x, y) = \frac{1}{N} \sqrt{\sum_{(x', y') \in D} |\theta(x', y') - \theta(x, y)|^2} \quad (3.53)$$

$$|\theta(x', y') - \theta(x, y)| = \begin{cases} d & \text{se } (d = (\theta(x', y') - \theta(x, y) + 360^\circ) \bmod 360^\circ) < 180^\circ \\ d - 180^\circ & \text{para todo o resto} \end{cases} \quad (3.54)$$

onde:

D – vizinhança local em torno do pixel (x, y) , o valor de D utilizado neste trabalho é 5×5 pixels.

N – número de pixels dentro de D .

$\bmod$ – operador resto da divisão de $(\theta(x', y') - \theta(x, y) + 360^\circ)$ por 360° .

$\theta(x', y')$ e $\theta(x, y)$ – orientações locais para os pixels (x', y') e (x, y) , respectivamente.

(x, y) – coordenada do pixel central da vizinhança D .

(x', y') – coordenada de um pixel pertencente a vizinhança D .

2º – Se $C(x, y)$ é menor do que um certo limiar T_C , o valor da orientação de campo para a posição (x, y) deve ser re-estimado utilizando-se a média das orientações de campo dos pixels pertencentes a região D . O valor de T_C varia dependendo da qualidade da imagem e do tamanho do bloco W . Neste trabalho foi utilizado um T_C igual a 25 (esse valor foi encontrado após utilizar vários valores de T_C nas imagens utilizadas nesta dissertação, pois os valores de TC dependem do brilho e do contraste da imagens).

3.5 – Determinação da Área de Interesse

A área de interesse de uma imagem de impressão digital é a região onde essa impressão está localizada. O método proposto por Jain e outros [1] para a determinação dos pixels pertencentes a essa área utiliza a variável *nível de certeza* ($CL(x, y)$) que é calculada utilizando-se a Equação (3.54).

$$CL(x, y) = \sqrt{\frac{1}{W \times W} \cdot \left(\frac{V_x(x, y)^2 + V_y(x, y)^2}{V_e(x, y)} \right)} \quad (3.55)$$

onde:

W – tamanho do bloco da vizinhança local na imagem.

$V_e(x, y)$ – definido pela Equação (3.56).

$$V_e(x, y) = \sum_{u=x-\frac{W}{2}}^{x+\frac{W}{2}} \sum_{v=y-\frac{W}{2}}^{y+\frac{W}{2}} \left[(G_x(u, v))^2 + (G_y(u, v))^2 \right] \quad (3.56)$$

Os passos para determinar se um pixel pertence a área de interesse ou não são:

1º – Calcula-se o *nível de certeza* para cada pixel da imagem $f(x, y)$; e

2º – Se o pixel tiver um nível de certeza menor do que um certo limiar, esse pixel é marcado como sendo um pixel de fundo. Caso contrário, não se altera o valor do nível de cinza desse pixel.

Por exemplo, para uma imagem com 256 níveis de cinza um pixel de fundo pode ser considerado como um pixel cujo nível de cinza é 255 (maior valor de nível de cinza possível nesse caso).

3.6 – Extração de Saliências

Considere $f(x, y)$ como sendo uma imagem de uma impressão digital. Para a extração das saliências nessa imagem, Jain e outros [1] propuseram que $f(x, y)$ seja convoluída com duas máscaras h_t e h_b , como definidas nas Equações (3.57) e (3.58), onde cada uma dessas máscaras, são convoluídas separadamente com a imagem $f(x, y)$, gerando duas novas imagens.

$$h_t(x, y; u, v) = \begin{cases} -\frac{1}{\sqrt{2\pi}\delta} e^{-\frac{u}{\delta^2}} & \text{se } u = v \cot[\alpha(x, y)] - \frac{H}{2 \cos[\alpha(x, y)]} \\ \frac{1}{\sqrt{2\pi}\delta} e^{-\frac{u}{\delta^2}} & \text{se } u = v \cot[\alpha(x, y)] \\ 0 & \text{para todo o resto} \end{cases} \quad (3.57)$$

$$h_b(x, y; u, v) = \begin{cases} -\frac{1}{\sqrt{2\pi\delta}} e^{-\frac{u}{\delta^2}} & \text{se } u = v \cot[\alpha(x, y)] + \frac{H}{2 \cos[\alpha(x, y)]} \\ \frac{1}{\sqrt{2\pi\delta}} e^{-\frac{u}{\delta^2}} & \text{se } u = v \cot[\alpha(x, y)] \\ 0 & \text{para todo o resto} \end{cases} \quad (3.58)$$

onde:

$$v \in \left[-\left\lceil \frac{L \sin[\alpha(x, y)]}{2} \right\rceil, \left\lceil \frac{L \sin[\alpha(x, y)]}{2} \right\rceil \right] \quad (3.59)$$

$$\alpha(x, y) = \theta(x, y) - 90^\circ.$$

H – número de linhas das máscaras h_t e h_b .

L – número de colunas das máscaras h_t e h_b .

δ – constante.

$\theta(x, y)$ – orientação de campo para o pixel na posição (x, y) .

O valor de δ é uma constante muito grande para melhorar a aproximação dos valores atribuídos às duas máscaras de convolução h_t e h_b . Neste trabalho δ foi utilizado com o valor igual a 25.000.

Para exemplificar o valor de $\delta = 25.000$, considere a Equação (3.60). Se δ é muito grande, o valor dessa equação tende a um, pois o argumento da função exponencial tende a zero. Logo, é proposto neste trabalho que as Equações (3.57) e (3.58) sejam aproximadas

utilizando-se as Equações (3.61) e (3.62), assim elimina-se o cálculo da função exponencial e com isso diminui-se o tempo computacional.

$$y = e^{-\frac{u}{\delta^2}} \quad (3.60)$$

$$h_t(x, y; u, v) = \begin{cases} -\frac{1}{\sqrt{2\pi\delta}} & \text{se } u = v \cot[\alpha(x, y)] - \frac{H}{2 \cos[\alpha(x, y)]} \\ \frac{1}{\sqrt{2\pi\delta}} & \text{se } u = v \cot[\alpha(x, y)] \\ 0 & \text{para todo o resto} \end{cases} \quad (3.61)$$

$$h_b(x, y; u, v) = \begin{cases} -\frac{1}{\sqrt{2\pi\delta}} & \text{se } u = v \cot[\alpha(x, y)] + \frac{H}{2 \cos[\alpha(x, y)]} \\ \frac{1}{\sqrt{2\pi\delta}} & \text{se } u = v \cot[\alpha(x, y)] \\ 0 & \text{para todo o resto} \end{cases} \quad (3.62)$$

O tamanho típico das máscaras de convolução proposto por [1] e utilizado neste trabalho é de 11×7 . A Figura 3.15 mostra as coordenadas das posições dessas máscaras.

(-5,-3)	(-5,-2)	(-5,-1)	(-5,0)	(-5,1)	(-5,2)	(-5,3)
(-4,-3)	(-4,-2)	(-4,-1)	(-4,0)	(-4,1)	(-4,2)	(-4,3)
(-3,-3)	(-3,-2)	(-3,-1)	(-3,0)	(-3,1)	(-3,2)	(-3,3)
(-2,-3)	(-2,-2)	(-2,-1)	(-2,0)	(-2,1)	(-2,2)	(-2,3)
(-1,-3)	(-1,-2)	(-1,-1)	(-1,0)	(-1,1)	(-1,2)	(-1,3)
(0,-3)	(0,-2)	(0,-1)	(0,0)	(0,1)	(0,2)	(0,3)
(1,-3)	(1,-2)	(1,-1)	(1,0)	(1,1)	(1,2)	(1,3)
(2,-3)	(2,-2)	(2,-1)	(2,0)	(2,1)	(2,2)	(2,3)
(3,-3)	(3,-2)	(3,-1)	(3,0)	(3,1)	(3,2)	(3,3)
(4,-3)	(4,-2)	(4,-1)	(4,0)	(4,1)	(4,2)	(4,3)
(5,-3)	(5,-2)	(5,-1)	(5,0)	(5,1)	(5,2)	(5,3)

Figura 3.15 – Coordenadas de uma máscara de convolução 11×7 utilizada neste trabalho.

O algoritmo para geração das máscaras de convolução h_t e h_b segue os seguintes passos:

1º – Determina-se os limites do intervalo que v pertence utilizando-se a Equação (3.59);

2º – Estabelece-se os valores inicial e final do intervalo de v , e;

3º – Aplica-se as Equações (3.61) e (3.62) para a atribuição dos valores dos coeficientes das máscaras h_t e h_b utilizando-se os valores inteiros contidos no intervalo de v .

Verifica-se quais valores obtidos de u estão dentro do intervalo $\left[-\frac{(H-1)}{2}, \frac{(H-1)}{2}\right]$,

esses valores são arredondados e atribuídos as suas respectivas posições.

Para todos os exemplos, é gerada somente a máscara h_t . O cálculo da máscara h_b utiliza o mesmo procedimento.

Considere que $\delta = 25.000$ e $\theta(x, y) = 60^\circ$. Primeiro, se $\theta(x, y) = 60^\circ$, então $\alpha(x, y) = -30^\circ$. O intervalo de v é:

$$v \in \left[-\left\lfloor \frac{L \sin(-30^\circ)}{2} \right\rfloor, \left\lfloor \frac{L \sin(30^\circ)}{2} \right\rfloor \right] \quad (3.63)$$

$$v \in [-1,75; 1,75] \quad (3.64)$$

u e v também indicam, respectivamente, a linha e a coluna de um elemento de uma das máscaras de convolução h_t ou h_b , portanto os valores de u e v são números inteiros. Então, quando um intervalo é não inteiro, como o mostrado na Equação (3.64), existe a necessidade de arredondá-lo para o valor inteiro mais próximo, como mostrado na Equação (3.65).

$$v \in [-2; 2] \quad (3.65)$$

Antes de aplicar o 3º passo do algoritmo para a geração das máscaras de convolução h_t e h_b , é necessário calcular a constante $c = \frac{1}{\sqrt{2\pi\delta}}$.

$$c = \frac{1}{\sqrt{2\pi \times 25.000}} \cong 2,5231 \cdot 10^{-3} \quad (3.66)$$

A Tabela 3.3 mostra o cálculo dos valores de u a partir de v no algoritmo para a geração das máscaras de convolução h_t e h_b .

Tabela 3.3 – Valores de u a partir de v no algoritmo para a geração das máscaras de convolução h_t e h_b .

v	$u = v \cot[\alpha(x, y)] - \frac{H}{2 \cos[\alpha(x, y)]}$	$u = v \cot[\alpha(x, y)]$
-2	-2,89	3,46
-1	-4,62	1,73
0	-6,35	0
1	-8,08	-1,73
2	-9,81	-3,46

Como mostrado na Tabela 3.3, para cada valor de v existem dois valores possíveis de u . Esses valores devem estar no intervalo $[-5; 5]$. Caso o valor de u não esteja nesse intervalo, a coordenada (u, v) é desconsiderada. Para exemplificar isso, considere nessa tabela quando v é igual a 2, onde a segunda coluna resulta em um valor de u igual a -9,81. Este valor não é considerado, pois está fora do intervalo permitido para u .

A partir dos valores de u e v obtidos e mostrados na Tabela 3.3 resulta a máscara h_t da Figura 3.16.

0	0	$-c$	0	0	0	0
0	0	0	0	0	0	0
0	$-c$	0	0	0	c	0
0	0	0	0	c	0	0
0	0	0	0	0	0	0
0	0	0	c	0	0	0
0	0	0	0	0	0	0
0	0	c	0	0	0	0
0	c	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

Figura 3.16 – Resultado da máscara h_t para $\theta(x, y) = 60^\circ$.

Por definição, antes de convoluir a máscara h_t e h_b é necessário realizar o rebatimento das mesmas. Então, a máscara h_t após o rebatimento é como mostrado na Figura 3.17.

0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	c	0
0	0	0	0	c	0	0
0	0	0	0	0	0	0
0	0	0	c	0	0	0
0	0	0	0	0	0	0
0	0	c	0	0	0	0
0	c	0	0	0	$-c$	0
0	0	0	0	0	0	0
0	0	0	0	$-c$	0	0

Figura 3.17 – Máscara h_t de tamanho 11×7 , $\alpha = -60^\circ$ e $\delta = 25.000$.

Depois de obtido o intervalo de v , cria-se um laço que deve gerar uma nova máscara h_t ou h_b para cada posição da imagem.

No caso da convolução de uma imagem $f(x, y)$ de dimensão $M \times N$ com uma máscara de dimensão $L \times H$, o resultado é uma imagem $g(x, y)$ de tamanho $M + L - 1 \times N + H - 1$. Essas novas $L - 1$ linhas e $H - 1$ colunas adicionais geradas pela convolução, podem ser desconsideradas. Isto pelo fato de que as características da imagem, não estão localizadas em suas bordas.

3.6.1 – Geração de máscaras de convolução mais suavizadas

Foi desenvolvido neste trabalho uma nova forma de obter as máscaras de convolução, h_t e h_b , resultando assim em imagens de melhor qualidade.

A melhor forma de determinar u e v é analisar as Equações (3.60) e (3.61).

Na Equação (3.61) a expressão $u = v \cot[\alpha(x, y)]$ é a equação de uma reta, onde v é a variável independente e $\cot[\alpha(x, y)]$ é o coeficiente angular. É importante notar que $\cot[\alpha(x, y)]$ é constante para a posição (x, y) .

Na Equação (3.62) $u = v \cot[\alpha(x, y)] + \frac{H}{2 \cos[\alpha(x, y)]}$ a fração é o coeficiente linear da

reta que também é constante para uma posição (x, y) .

Neste trabalho, para melhorar os resultados de u e v nas máscaras de convolução h_t e h_b , utiliza-se o algoritmo Digital Differential Analyzer (DDA) [18] que é usado na geração das coordenadas dos pontos das retas.

3.6.2 – Algoritmo Digital Differential Analyzer (DDA)

O algoritmo DDA foi proposto por Hearn e Baker [18]. Ele é utilizado para melhorar a alocação discreta da coordenada (x, y) de uma reta. Esse algoritmo foi desenvolvido porque por exemplo, em monitores de computadores, é possível representar somente valores inteiros das coordenadas (x, y) . Como uma reta possui infinitos pontos, para alguns casos o resultado visual é de uma reta com falhas. Para resolver esse problema, considere a Equação (3.67) como a equação de uma reta.

$$y = x \cdot m + n \quad (3.67)$$

$$m = \tan \alpha \quad (3.68)$$

onde:

m – coeficiente angular da reta.

n – coeficiente linear da reta.

α – ângulo entre a reta e o eixo x .

A Figura 3.18 mostra o ângulo α formado entre a reta e o eixo x .

O valor do coeficiente angular m pode ser obtido a partir da Equação (3.68) ou da Equação (3.69).

$$m = \frac{\Delta y}{\Delta x} = \frac{y_B - y_A}{x_B - x_A} \quad (3.69)$$

onde:

Δy – taxa de variação do valor da ordenada y entre os dois pontos A e B da Equação (3.67).

Δx – taxa de variação do valor da abscissa x entre os dois pontos A e B da Equação (3.67).

x_A e y_A – abscissa e ordenada do ponto A , da reta da Equação (3.67), respectivamente.

x_B e y_B – abscissa e ordenada do ponto B , da reta da Equação (3.67), respectivamente.

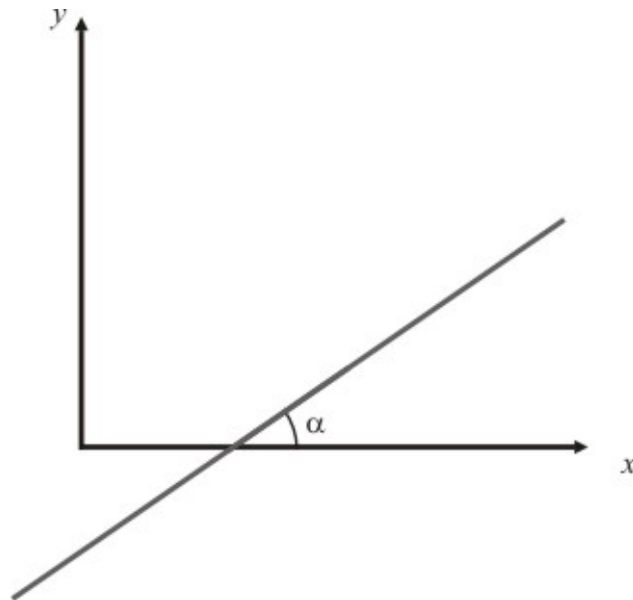


Figura 3.18 – Ângulo α formado entre a reta da Equação (3.66) e o eixo x .

Na Equação (3.69) quando $|\alpha| < 45^\circ$, a taxa de variação da variável x é maior do que a taxa de variação da variável y , ou seja, $\Delta x > \Delta y$. Então, ocorre um número maior de valores inteiros para x do que para y . Neste caso, para uma melhor determinação dos pontos da reta, utiliza-se a Equação (3.67) substituindo-se valores inteiros para x e determinando-se os valores de y . Caso na Equação $|\alpha| > 45^\circ$, y tem uma maior taxa de variação do que x , então é

obtido um melhor resultado utilizando-se a Equação (3.70), substituindo-se valores inteiros para y determinando-se os valores de x . Quando $\theta = 45^\circ$, $\Delta x = \Delta y$, utiliza-se a Equação (3.67) para calcular o valor de x .

$$x = \frac{1}{m}(y - n) \quad (3.70)$$

O algoritmo DDA pode ser aplicado as equações que determinam as coordenadas das máscaras de convolução h_t e h_b . A determinação das posições dessas máscaras é utilizada da seguinte maneira: se $|\alpha|$ é menor ou igual a 45° , utilizam-se as Equações (3.61) e (3.62) para gerar essas máscaras. Caso contrário, essas equações são rearranjadas de forma a obter as Equações (3.71) e (3.72).

$$h_t(x, y; u, v) = \begin{cases} -\frac{1}{\sqrt{2\pi\delta}} & \text{se } v = u \tan[\alpha(x, y)] + \frac{H}{2 \cos[\alpha(x, y)]} \\ \frac{1}{\sqrt{2\pi\delta}} & \text{se } v = u \tan[\alpha(x, y)] \\ 0 & \text{para todo o resto} \end{cases} \quad (3.71)$$

$$h_b(x, y; u, v) = \begin{cases} -\frac{1}{\sqrt{2\pi\delta}} & \text{se } v = u \tan[\alpha(x, y)] - \frac{H}{2 \cos[\alpha(x, y)]} \\ \frac{1}{\sqrt{2\pi\delta}} & \text{se } v = u \tan[\alpha(x, y)] \\ 0 & \text{para todo o resto} \end{cases} \quad (3.72)$$

3.7 – Afinamento das Saliências de Impressões Digitais

Depois de convoluir a imagem da impressão digital com as duas máscaras de convolução h_t e h_b e obter a imagem binarizada, é necessário aplicar nessa imagem o algoritmo de *afinamento* (ou *esqueletização*) para transformar as saliências em linhas com espessura de um pixel.

Como definido por Gonzalez e Woods [15] a representação estrutural da forma de uma região planar consiste em reduzi-la a um *grafo*. Essa redução é realizada obtendo-se o *esqueleto* da região.

As regiões binárias de uma imagem de impressão digital que passaram por um processo de afinamento, ou seja, as saliências, possuem pixels com valor igual a 1, e os pixels de fundo possuem valor igual a 0.

O algoritmo de afinamento proposto por Gonzalez e Woods [15] e utilizando neste trabalho consiste na aplicação sucessiva de dois passos aos pixels do contorno da região. Um pixel do contorno é definido como um pixel de valor igual a 1 que possui pelo menos um pixel vizinho de 8 com valor igual a 0. Considere a vizinhança de oito, como mostrado na Figura 3.19.

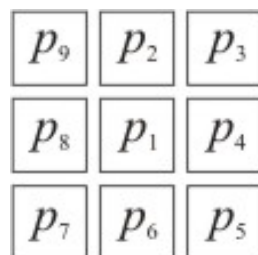


Figura 3.19 – Vizinhança de oito do pixel p_1 .

O primeiro passo do algoritmo de afinamento proposto por Gonzalez e Woods [15] determina que um pixel do contorno p , deve ser eliminado se as seguintes condições forem satisfeitas:

$$\begin{aligned}
 & \text{(a) } 2 \leq N(p_1) \leq 6; \\
 & \text{(b) } S(p_1) = 1; \\
 & \text{(c) } p_2 \cdot p_4 \cdot p_6 = 0; \text{ e} \\
 & \text{(d) } p_4 \cdot p_6 \cdot p_8 = 0.
 \end{aligned} \tag{3.73}$$

onde:

$N(p_1)$ – número de pixels vizinhos não-nulos de p_1 , ou seja:

$$N(p_1) = p_2 + p_3 + p_4 + p_5 + p_6 + p_7 + p_8 + p_9 \tag{3.74}$$

$S(p_1)$ – número de transições de 0 para 1 na seqüência $p_2, p_3, \dots, p_8, p_9, p_2$.

A Figura 3.20 mostra uma vizinhança de oito, onde $N(p_1) = 4$ e $S(p_1) = 3$.

0	1	0
0	p_1	1
1	1	0

Figura 3.20 – Vizinhança de oito, onde $N(p_1) = 4$ e $S(p_1) = 3$.

O segundo passo do algoritmo de afinamento proposto por Gonzalez e Woods [15] possui as seguintes condições:

$$\begin{aligned}
& \text{(a) } 2 \leq N(p_1) \leq 6; \\
& \text{(b) } S(p_1) = 1; \\
& \text{(c) } p_2 \cdot p_4 \cdot p_8 = 0; \text{ e} \\
& \text{(d) } p_2 \cdot p_6 \cdot p_8 = 0.
\end{aligned} \tag{3.75}$$

O primeiro passo do algoritmo de afinamento proposto por Gonzalez e Woods [15] deve ser aplicado a cada pixel da borda da região binária. Se uma das condições das letras (a) a (d) da Equação (3.73) desse passo forem violadas, o valor do pixel em questão não deve ser mudado. Se todas as condições do primeiro passo forem satisfeitas, o valor do pixel é marcado para ser apagado. O pixel não deve ser apagado até que todos os pixels da borda tenham sido analisados. Uma vez que todos eles tenham sido analisados, os que foram marcados devem ser apagados, ou seja, atribuídos a estes pixels o valor 0. Em seguida, aplica-se o segundo passo aos pixels da imagem resultante da primeira iteração. Depois da aplicação do segundo passo e apagados os pixels que foram marcados pelo algoritmo, o primeiro e o segundo passos são realizados novamente. Quando em uma iteração desse algoritmo não houver mais pixels a serem apagados, a imagem obtida é o esqueleto da região binária.

A Figura 3.21 mostra a execução do primeiro e do segundo passos do algoritmo de afinamento desenvolvido por Gonzalez e Woods [15] e o resultado final desse algoritmo.

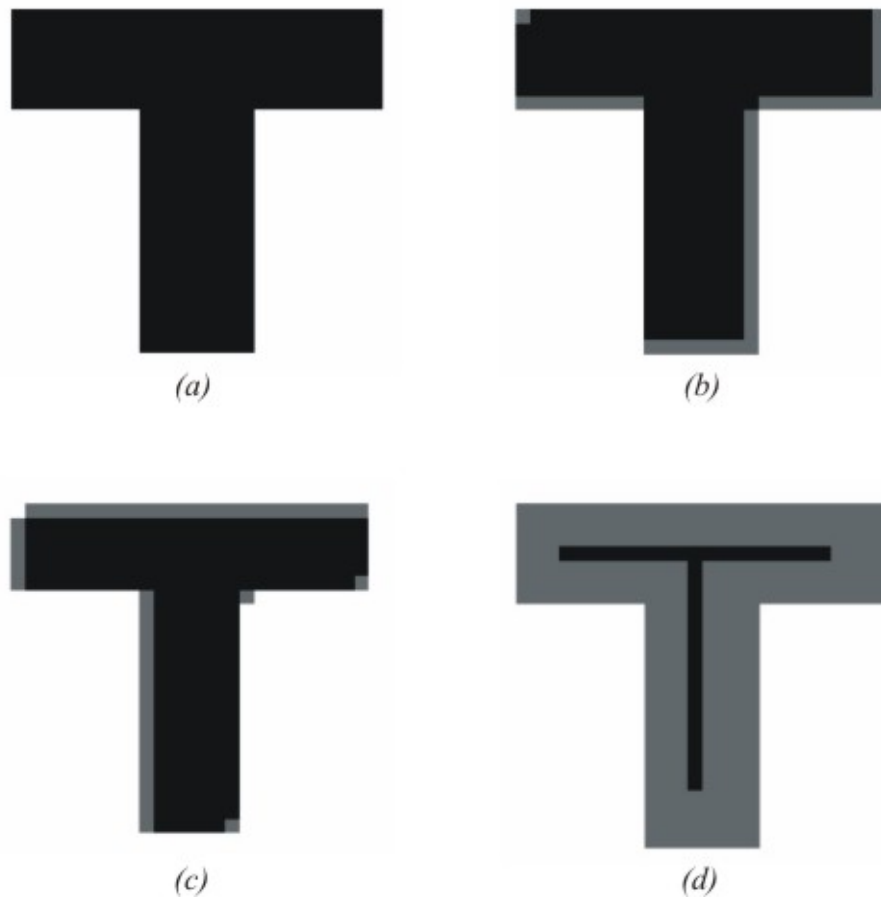


Figura 3.21 – Afinamento de uma imagem. (a) imagem original; (b) pixels eliminados pelo 1º passo; (c) pixels eliminados pelo 2º passo; (d) imagem resultante do afinamento com os pixels eliminados.

As Figuras 3.21(b) e (c) mostram na cor cinza, os pixels apagados na primeira iteração pelo primeiro e pelo segundo passos, respectivamente. A Figura 3.21(d) mostra a imagem resultante (representada pelos pixels na cor preta) após não existirem mais pixels a serem apagados. Os pixels da cor cinza na Figura 3.21(d) são os pixels apagados pelo o algoritmo de afinamento proposto por Gonzalez e Woods [15].

As Figuras 3.22(a) e (b) mostram uma imagem binária de uma impressão digital e o resultado final do algoritmo de afinamento proposto por Gonzalez e Woods [15] aplicado a imagem da Figura 3.22(a), respectivamente.



Figura 3.22 – (a) Imagem original; (b) imagem afinada pelo o algoritmo de afinamento.

Existem alguns pixels da imagem resultante do algoritmo de afinamento proposto por Gonzalez e Woods [15] que precisam serem eliminados, pois resultam na detecção de falsas minúcias. Foi desenvolvido neste trabalho um algoritmo para eliminação desse pixels residuais. Esse algoritmo consiste em comparar a vizinhança de oito de cada pixel da imagem com as vizinhanças de oito mostradas na Figura 3.23. Se vizinhança de oito de um pixel coincidir com qualquer uma das vizinhanças mostradas na Figura 3.23, então o pixel central é apagado. É importante ressaltar que os pixels das saliências são marcados com o valor igual a 1, enquanto os pixels de fundo são marcados com o valor igual a 0. Nas imagens dessa figura, os pixels marcados com X podem possuir o valor igual a 0 ou 1, não importa.

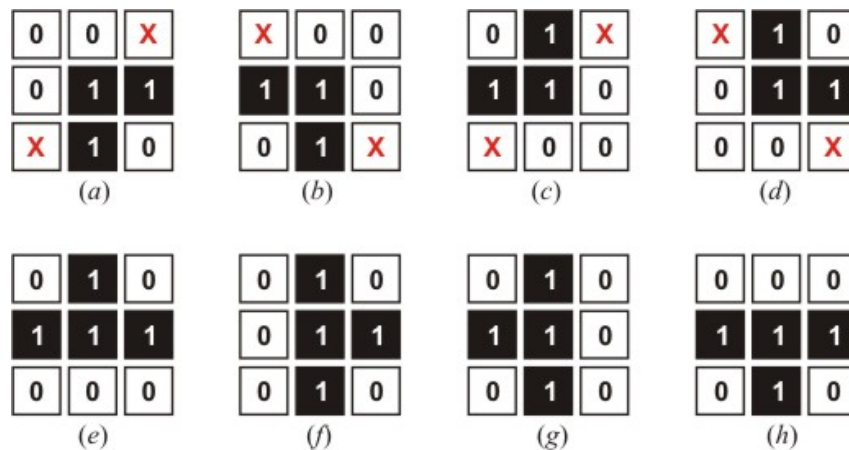


Figura 3.23 – Vizinhanças de oito dos pixels que geram falsas minúcias.

3.8 – Detecção de Minúcias

Obtido o esqueleto da imagem da impressão digital, Jain e outros [1] propuseram que um pixel é classificado como minúcia do tipo terminação de saliência se $N(p_1) = 1$, onde $N(p_1)$ é o número de pixels vizinhos como definido na Equação (3.74). Um pixel é classificado como bifurcação de saliência se $N(p_1) > 2$. A Figura 3.24 mostra um exemplo de minúcias dos tipos terminação e bifurcação de saliência. As Figuras 3.24(a) e (b) mostram os pixels, que estão realçados por um quadrado vermelho, como uma minúcia do tipo terminação e bifurcação, respectivamente.

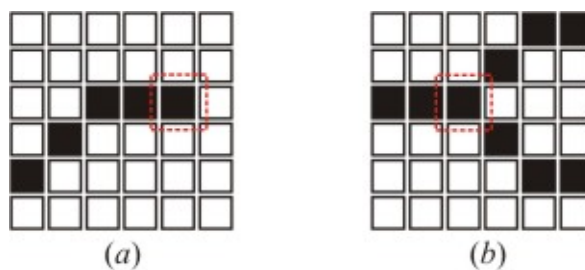


Figura 3.24 – (a) Minúcia do tipo terminação; (b) minúcia do tipo bifurcação.

Quando é detectada a minúcia do tipo bifurcação com $N(p_1) = 3$, essa minúcia possui duas falsas minúcias (que estão representadas pela cor vermelha na Figura 3.25). Para eliminar essa detecção errônea foi desenvolvido neste trabalho um algoritmo que compara a vizinhança de oito do pixel que é classificado como minúcia quando $N(p_1) = 3$ com as vizinhanças de oito mostradas na Figura 3.26. Se a vizinhança do pixel ocorrer em algum dos casos da Figura 3.26, então o pixel é classificado como minúcia do tipo bifurcação. Caso contrário o pixel é um pixel de saliência.

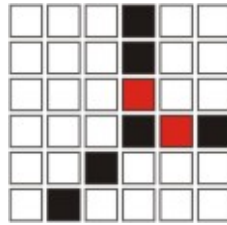


Figura 3.25 – Exemplo de duas falsas minúcias.

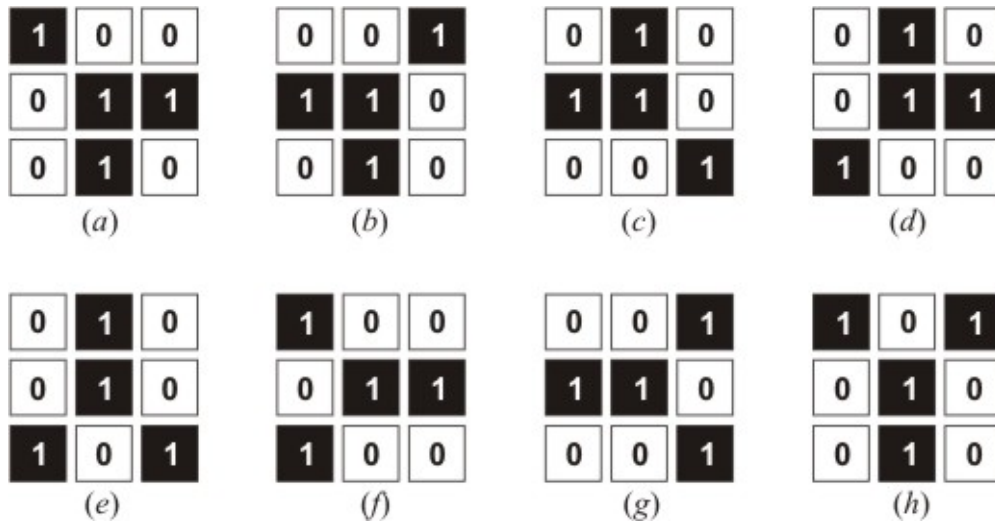


Figura 3.26 – Vizinhanças de oito para minúcias do tipo bifurcação com $N(p_1) = 3$.

Na obtenção da imagem de uma impressão digital podem ocorrer imperfeições, por exemplo, devido a doenças de pele, bolhas nos dedos, ferimentos, suor, entre outras causas. Devido a isso, existe o aparecimento de espinhos, cortes e junções de saliências na imagem afinada dessa impressão que ocasiona a detecção de minúcias espúrias que interferem no reconhecimento correto.

Como proposto por Jain e outros [1], para cada minúcia detectada são armazenados os seguintes parâmetros: abscissa x , ordenada y , orientação de campo e o segmento da saliência associado a ela. As saliências armazenadas são representadas como sinais discretos unidimensionais normalizados utilizando-se como referência a distância média entre elas. Essa normalização garante a flexibilidade do algoritmo trabalhar com imagens de tamanhos diferentes de uma mesma impressão digital. Por exemplo, considere uma mesma impressão digital representada por duas imagens de tamanhos 128×128 pixels e 256×256 pixels; e dessas imagens são extraídos dois sinais unidimensionais que representam a mesma saliência. Se entre esses dois sinais a comparação é realizada sem normalização, não é identificada a semelhança entre eles, pois um é o dobro do outro. A normalização elimina esse problema.

3.9 – Correção de Falhas Presentes em Saliências de Uma Imagem de Impressão Digital

Após a binarização e o processo de afinamento é comum aparecer falhas nas saliências de uma imagem de impressão digital, gerando assim minúcias espúrias que podem dificultar a etapa de reconhecimento. Com o aparecimento dessas minúcias ocorrem pixels comuns como se fossem minúcias do tipo terminação. Neste trabalho foi desenvolvido um algoritmo para corrigir essas falhas, onde para cada pixel classificado como minúcia do tipo terminação,

tenta-se encontra um outro pixel que se enquadre nos requisitos para que esse pixel possa ser ligado ao pixel dessa minúcia. Os passos do algoritmo desenvolvido neste trabalho para eliminação dessas falhas são:

- 1º - Quando um pixel é classificado como minúcia do tipo terminação, ir para o segundo passo, caso contrário, o pixel em questão é uma minúcia do tipo terminação;
- 2º - Testar se existe um outro pixel que possa ser ligado ao pixel da minúcia do tipo terminação. Os possíveis pixels que podem ser ligados são delimitados por uma região de teste (esta região será explicada a seguir neste mesmo item deste capítulo);
- 3º - Se é encontrado algum pixel com valor diferente de zero (pixels com valores iguais a zero são pixels de vale) dentro da região de teste, a vizinhança de oito desse pixel é testada.
- 4º - Se o pixel encontrado na região de teste possui um único pixel vizinho de valor igual a 1, e o sentido (da esquerda para a direita ou da direita para a esquerda) da saliência desse pixel é oposto ao sentido da saliência do pixel classificado como minúcia do tipo terminação, esses dois pixels são ligados eliminando a falha na saliência. Caso alguma das condições não seja satisfeita, o pixel encontrado dentro da região de teste não é ligado ao pixel classificado como pixel de minúcia do tipo terminação.

A Figura 3.27 mostra o fluxograma do algoritmo desenvolvido neste trabalho para a eliminação de falhas em saliências.

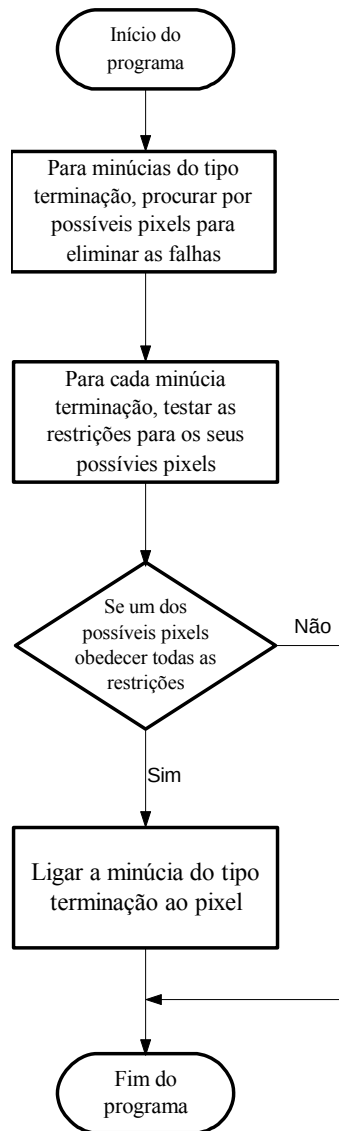


Figura 3.27 – Fluxograma do algoritmo para eliminação de falhas em saliência.

A Figura 3.28 mostra a execução do algoritmo desenvolvido neste trabalho para eliminação de falhas em uma saliência.

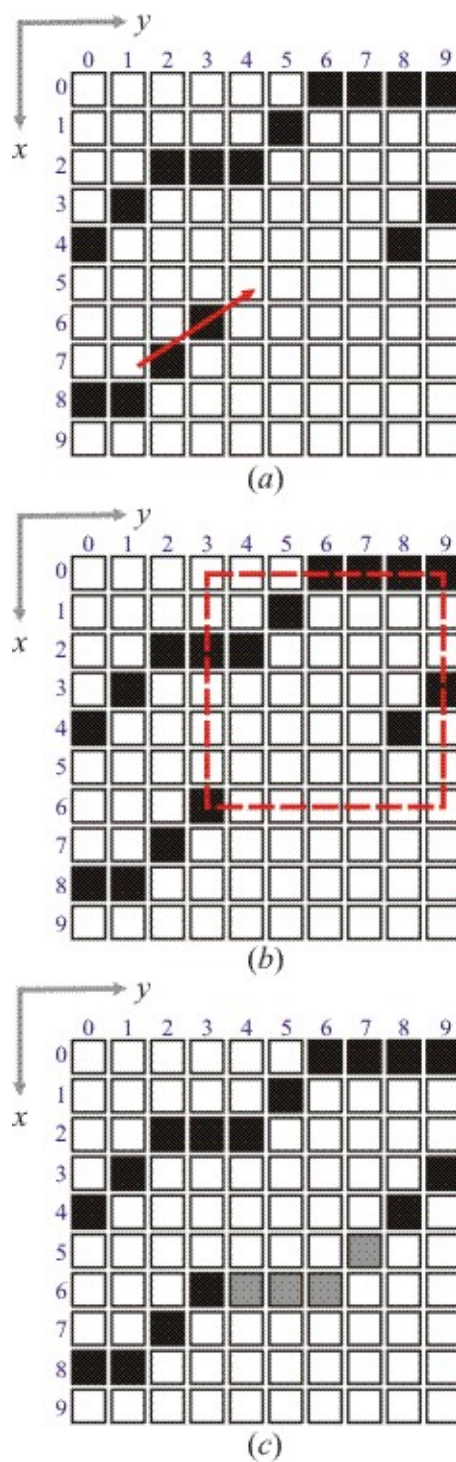


Figura 3.28 – Exemplo de eliminação de uma falha em uma saliência.

A Figura 3.28(a) mostra o pixel de coordenada (6, 3) classificado como uma minúcia do tipo terminação. A direção da orientação de campo desse pixel é mostrada pela seta de cor vermelha.

A Figura 3.28(b) mostra a região de teste delimitada pela linha na cor vermelha. Alguns pixels da saliência acima da saliência que está sendo aplicado o algoritmo desenvolvido neste trabalho para eliminação de falhas em saliências podem ser os possíveis pixels necessários para ligar as saliências. Mas as condições sobre a vizinhança do pixel não são satisfeitas, logo esse algoritmo testa os outros pixels até encontrar o pixel de coordenada igual a (4, 8) onde esse pixel possui somente um pixel vizinho.

O último teste a ser realizado na Figura 3.28(b) no pixel de coordenada igual a (4, 8) é se o sentido desse pixel está concordando com o sentido do pixel considerado como minúcia do tipo terminação. Se isso ocorrer, o pixel considerado como minúcia possui uma saliência que vem da esquerda para a direita. O pixel de coordenada (4, 8) possui uma saliência que vai da direita para a esquerda, concordando com o sentido do pixels considerado como minúcia do tipo terminação, logo esses dois pixels devem ser ligados eliminando assim a falha na saliência, como mostrada a Figura 3.28(c), onde os pixels na cor cinza são os novos pixels da saliência.

As coordenadas dos novos pixels da saliência podem ser determinadas utilizando-se o algoritmo DDA.

Uma outra restrição aplicada aos possíveis pixels necessários para ligar as saliências para a remoção das falhas nelas é a distância entre a minúcia e esses possíveis pixels. Como proposto por Jain e outros [1], essa distância tem de ser menor do que 15 pixels. Depois de

determinadas as coordenadas da minúcia e desses possíveis pixels, a distância entre esses dois pontos é calculada utilizando-se a Equação (2.6) da Seção 2.4 deste trabalho.

3.10 – Determinação das Coordenadas dos Pixels de uma Saliência

Por exemplo, no reconhecimento de impressão digital, às vezes é necessário percorrer uma saliência por inteiro para determinar seu tamanho. A função do algoritmo de determinação das coordenadas dos pixels de uma saliência é determinar os pixels pertencentes a ela. Foi desenvolvido neste trabalho um algoritmo que a partir da coordenada do primeiro pixel de uma saliência, determina-se as coordenadas de todos os outros pixels dessa saliência.

Para o algoritmo de determinação das coordenadas dos pixels de uma saliência, considere a Figura 3.29 e os valores dos pixels pertencentes a saliência iguais a 1 e os valores dos pixels de fundo iguais a zero. Quando é implementado o teste dos vizinhos de oito do pixel de uma saliência para decidir qual é o próximo pixel, normalmente esse teste é implementado utilizando-se um laço ou mesmo um conjunto de *if's*. É muito importante observar que normalmente o teste sempre começa por uma mesma posição, por exemplo p_2 , e se a posição não é a posição do próximo pixel testa-se as outras posições até encontrar o pixel ou completar o teste para as oito posições. Para exemplificar, considere a Figura 3.29, onde o pixel vermelho é a minúcia. A posição seguinte é a de coordenada $(x + 1, y - 1)$ (pixel p_7) em relação a atual. Por exemplo, quando é realizado o teste para o próximo pixel para a nova posição e esse teste segue uma ordem pré-definida na sequência de p_2 até p_9 , então quem é considerado o próximo pixel é a minúcia. Desse modo, o teste do próximo pixel fica mudando de posição entre a minúcia e o pixel posterior a ela infinitamente por causa da sequência desse

procedimento. Para evitar isso deve-se marcar os pixels já testados com algum valor que evita esse erro (por exemplo, neste trabalho utilizou-se o valor de -1).

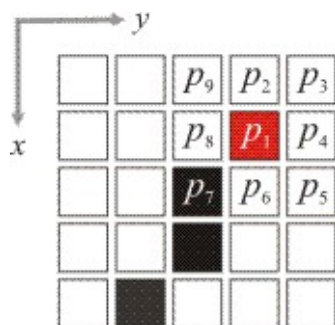


Figura 3.29 – Teste do próximo pixel.

As coordenadas dos pixels vizinhos ao pixel p_1 da Figura 3.29 são mostradas na Tabela 3.4.

Tabela 3.4 – Coordenadas dos pixels vizinhos ao pixel p_1 da Figura 3.29

<i>Pixel</i>	<i>Coordenada</i>
p_2	$(x - 1, y)$
p_3	$(x - 1, y + 1)$
p_4	$(x, y + 1)$
p_5	$(x + 1, y + 1)$
p_6	$(x + 1, y)$
p_7	$(x + 1, y - 1)$
p_8	$(x, y - 1)$
p_9	$(x - 1, y - 1)$

3.11 – Conversão das Saliências em Sinais Discretos Unidimensionais

Com a finalidade de auxiliar no reconhecimento de impressões digitais, depois que as minúcias são detectadas é necessário converter as saliências que estão associadas a elas em sinais discretos unidimensionais com a mesma direção da minúcia. O algoritmo desenvolvido nesta dissertação realiza a translação de todos os pixels de cada saliência de $-x_o$ em relação a x e $-y_o$ em relação a y , onde (x_o, y_o) é a coordenada da minúcia associada a saliência. Depois disso, aplica-se a rotação nos pixels translados.

A translação de um pixel de uma imagem é representada pela multiplicação de matrizes, como mostrado na Equação (3.76) [15].

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & x_o \\ 0 & 1 & 0 & y_o \\ 0 & 0 & 1 & z_o \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (3.76)$$

onde:

x, y e z – coordenada do pixel da saliência.

x', y' e z' – nova coordenada do pixel da saliência transladado.

x_o, y_o e z_o – distância em x, y e z , que o pixel é translado, respectivamente.

Realizada a multiplicação das matrizes na Equação (3.76), a Equação (3.77) é a nova coordenada do pixel p transladado.

$$p = \begin{bmatrix} x - x_o \\ y - y_o \\ z - z_o \\ 1 \end{bmatrix} \quad (3.77)$$

A rotação de um pixel na forma matricial é obtida pela Equação (3.78) [15].

$$p' = \begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} \quad (3.78)$$

onde:

θ – ângulo de rotação.

x' , y' e z' – coordenada do pixel de coordenada (x, y, z) transladado.

O valor do ângulo de rotação θ é igual a orientação de campo da minúcia, esse ângulo é positivo quando a rotação é no sentido horário.

As Equações (3.77) e (3.78) transladam e rotacionam de θ um pixel de coordenada (x, y, z) , respectivamente. Para obter as duas transformações geométricas é necessário substituir a Equação (3.77) na Equação (3.78), obtendo-se a Equação (3.79).

$$v = \begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x - x_o \\ y - y_o \\ z - z_o \\ 1 \end{bmatrix} \quad (3.79)$$

onde:

v – matriz que representa a nova coordenada do pixel transladado e rotacionado de θ .

Multiplicando-se as matrizes da Equação (3.79), obtém-se a Equação (3.80). Esta equação é a posição final do pixel de coordenada (x, y, z) após ele ter sido transladado e rotacionado de θ .

$$v = \begin{bmatrix} (x - x_o) \cos \theta + (y - y_o) \sin \theta \\ -(x - x_o) \sin \theta + (y - y_o) \cos \theta \\ z - z_o \\ 1 \end{bmatrix} \quad (3.80)$$

Como qualquer pixel da imagem encontra-se no plano xy e não existe nenhuma mudança no valor de z , então a expressão $z - z_o$ da Equação (3.80) pode ser desconsiderada, pois ela não afeta em nada a conversão da saliência em um sinal unidimensional. Assim, x' (abscissa) e y' (ordenada) são calculadas pelas Equações (3.81) e (3.82), respectivamente.

$$x' = (x - x_o) \cos \theta + (y - y_o) \sin \theta \quad (3.81)$$

$$y' = -(x - x_o) \sin \theta + (y - y_o) \cos \theta \quad (3.82)$$

onde:

x' e y' – novos valores da abscissa e da ordenada, do pixel da saliência, respectivamente.

x_o e y_o – abscissa e ordenada, do pixel classificado como minúcia, respectivamente.

Para exemplificar o uso das Equações (3.81) e (3.82), considere a saliência mostrada na Figura 3.30. A minúcia está realçada pelo pixel de cor cinza claro e o ângulo formado pela sua direção (indicado nessa figura pela reta r) com o eixo x é igual a 135° . Substituindo-se a coordenada da minúcia e o valor de θ nessas equações, obtém-se as Equações (3.83) e (3.84).

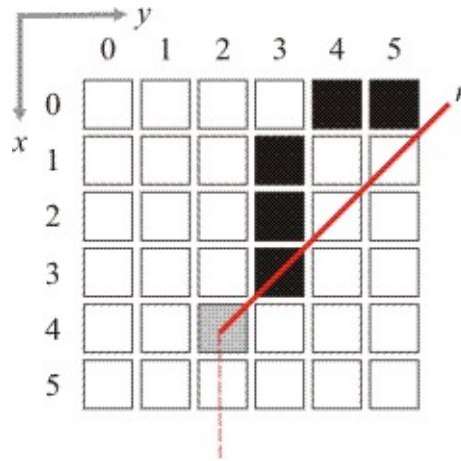


Figura 3.30 – Saliência com a minúcia de coordenada (4, 2).

$$x' = (x - 4) \cos (135^\circ) + (y - 2) \sin (135^\circ) \quad (3.83)$$

$$y' = -(x - 4) \sin (135^\circ) + (y - 2) \cos (135^\circ) \quad (3.84)$$

A Tabela 3.5 mostra as coordenadas originais (x, y) e as novas coordenadas (x', y') dos seis pixels pertencentes a saliência mostrada na Figura 3.30. Essas novas coordenadas são resultantes da aplicação da translação e da rotação nos pixels da saliência. No algoritmo implementado neste trabalho considera-se que o 1º pixel sempre é a minúcia associada à saliência, ou seja, nessa figura é o pixel de cor cinza claro de coordenada (4, 2).

Tabela 3.5 – Coordenadas originais e resultantes da translação e da rotação dos pixels pertencentes a saliência da Figura 3.30.

Pixel	$(x; y)$	$(x'; y')$
1º	(4; 2)	(0; 0)
2º	(3; 3)	(1,4142; 0)
3º	(2; 3)	(2,1213; 0.7071)
4º	(1; 3)	(2,8284; 1.4142)
5º	(0; 4)	(4,2426; 1.4142)
6º	(0; 5)	(4,9497; 0.7071)

Em um sinal discreto unidimensional, a variável x assume somente valores inteiros. Os valores das abscissas x dos pixels pertencentes as saliências resultantes da translação e da rotação são valores reais, como pode ser observado na Tabela 3.5. Uma forma de ajustar os valores dessas saliências é arredondar o valor de cada x para o inteiro mais próximo. A solução mais precisa, desenvolvida nesta dissertação, é encontrar novos valores de x e y do sinal unidimensional discreto utilizando-se a interpolação linear. Foi escolhido esse tipo de interpolação porque as saliências não se comportam como nenhuma função matemática, como por exemplo, função quadrática ou exponencial, logarítmica, trigonométricas, etc..

Para exemplificar a interpolação linear, considere um valor de $x \in [a, b]$, onde a e b são números reais e $f(x)$ é uma função que não apresenta descontinuidade nesse intervalo de x . Então, conclui-se que $f(a) \leq y \leq f(b)$, como mostra a Figura 3.31, onde $y = f(x)$.

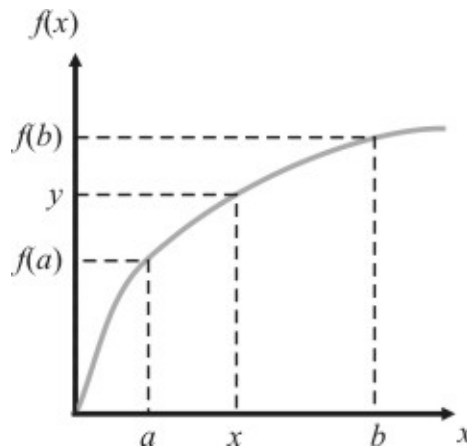


Figura 3.31 – Exemplo de uma função contínua em um intervalo $[a, b]$.

Conhecendo-se os valores de a , b , $f(a)$ e $f(b)$, pela interpolação linear determina-se os valores de x e y utilizando-se a Equação (3.85).

$$\frac{f(b) - f(a)}{b - a} = \frac{f(x) - f(a)}{x - a} \quad (3.85)$$

Rearranjando a Equação (3.85) e substituindo nela $f(x)$ por y , o valor de y é determinado pela Equação (3.86).

$$y = \left[\frac{f(b) - f(a)}{b - a} \right] \cdot (x - a) + f(a) \quad (3.86)$$

A interpolação linear não abrange valores que estão fora da faixa do intervalo de valores conhecidos. Na Tabela 3.5, o intervalo de valores de x é de zero a 4,9497, ou seja, somente valores dentro desse intervalo são estimados por essa interpolação. Como os valores

de x são valores inteiros e o maior valor inteiro que pode ser estimado nessa interpolação é 4. Então x pode ter como valores somente 0, 1, 2, 3 e 4.

Para determinar um valor de x utilizando-se a Equação (3.86), os valores de a e b são escolhidos de forma que esses valores (foram os valores obtidos da translação e rotação) sejam os valores mais próximos o possível do x a ser estimado.

Para exemplificar o uso da Equação (3.86), considere o valor de $f(2)$ utilizando-se os valores da Tabela 3.5. Os valores de a e b utilizados para encontrar $f(2)$, são respectivamente 1,4142 e 2,1213, por serem números reais obtidos da translação e da rotação mais próximos de 2. Substituindo-se os valores de a , b e $x = 2$, na Equação (3.85), obtém-se as Equações (3.87) e (3.88).

$$y = \left[\frac{f(2,1213) - f(1,4142)}{2,1213 - 1,4142} \right] \cdot (2 - 1,4142) + f(1,4142) \quad (3.87)$$

$$y = \left(\frac{0,7071 - 0}{2,1213 - 1,4142} \right) \cdot (2 - 1,4142) + 0 = 0,5858 \quad (3.88)$$

Pela Equação (3.88), para o valor de x igual a 2 o y estimado é igual a 0,5858. Todos os valores da Tabela 3.5 encontrados utilizando-se interpolação linear são mostrados na Tabela 3.6.

Tabela 3.6 – Novas coordenadas arredondadas dos pixels da Figura 3.30.

Pixel	$(x'; y')$
1º	(0; 0)
2º	(1; 0)
3º	(2; 0,5858)
4º	(3; 1,4142)
5º	(4; 1,4142)

O sinal unidimensional resultante da saliência mostrada na Figura 3.30 é mostrado na Figura 3.31.

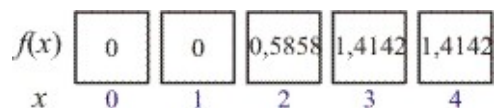


Figura 3.32 – Sinal unidimensional resultante da saliência mostrada na Figura 3.30

Como a imagem é digital, ou seja, os pixels ocupam posições discretas (valores inteiros), essa característica provoca erros de arredondamento na coordenada (x, y) de cada pixels. Neste trabalho, os processos de binarização e afinamento da imagem provocam erros na coordenada (x, y) de cada pixel. A interpolação é aplicada no processo de conversão das saliências em um sinal discreto unidimensional para amenizar esses erros associados as coordenadas (x, y) dos pixels da imagem.

O algoritmo de conversão de uma saliência em um sinal discreto unidimensional descrito anteriormente neste capítulo é para a saliência que começa em uma minúcia do tipo terminação. Para cada minúcia do tipo bifurcação, existem três possíveis saliências que podem ser convertidas em um sinal unidimensional. Na Figura 3.33, a cor vermelha mostra o

pixel classificado como minúcia do tipo bifurcação. No algoritmo desenvolvido neste trabalho para a escolha a saliência que será associada a minúcia do tipo bifurcação, propõe-se a convenção de percorrer a saliência no sentido que exista somente um possível caminho a ser percorrido, ou seja, para uma minúcia, existem dois possíveis caminhos nas direções de θ e $\theta + \pi$, onde θ é o valor da orientação de campo para a minúcia. Como mostrado nessa figura, a minúcia com a mesma direção do eixo x , possui dois sentidos possíveis para a conversão da saliência em um sinal unidimensional, ou ela é para cima ou ela é para baixo. Se o sentido escolhido é para cima, dois possíveis pixels podem ser escolhidos que são os pixels nas posições (2, 2) ou (2, 4). Pela convenção adotada neste trabalho, o sentido escolhido é para baixo, pois cada posição do sinal unidimensional discreto possui somente um caminho a ser percorrido, logo o pixel escolhido é o pixel de posição (3, 3).

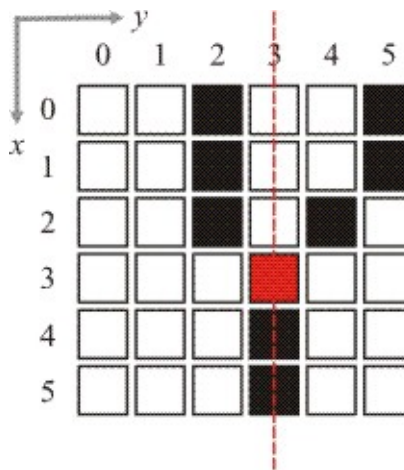


Figura 3.33 – Direção de uma minúcia do tipo bifurcação.

Os passos do algoritmo para determinar qual saliência será associada a minúcia do tipo bifurcação são:

- 1º - Determina-se as coordenadas dos três possíveis pixels, onde um desses pixels é o segundo pixel da saliência que será associada a minúcia do tipo bifurcação;
- 2º - Depois de determinado os três possíveis pixels no 1º passo são atribuídos a esses pixels o valor a zero (pixels com valor igual a 1 são pixels pertencentes a saliência), e então é percorrido n pixels em cada saliência, utilizando-se o algoritmo descrito na Seção 3.9 deste capítulo (algoritmo para eliminação de falhas presentes em saliências). O valor de n é igual a quantidade de pixels percorridos em uma das saliências. Esse valor tem um valor máximo que neste trabalho é de 10 pixels, ou atinge o limite de uma saliência que pode ser uma minúcia (tipo terminação ou bifurcação) ou é o limite da imagem que pode ser menor do que 10. São chamados de *extremos* de cada saliência os n -ésimos pixels dela;
- 3º - Entre os três extremos de cada saliência, existem três distâncias entre eles. Calcula-se essas distâncias (distância Euclidiana), como mostrado na Seção 2.4 do capítulo 2 deste trabalho;
- 4º - Entre três extremos existem três distâncias que podem ser calculadas. A menor distância encontrada foi determinada a partir dos dois extremos mais próximos, ou seja, as saliências que possuem esses dois extremos estão na direção que possui possíveis caminhos a serem percorridos, logo a saliência que será associada a minúcia é oposta a essa direção, ou seja, a saliência que contém o pixel que não foi utilizado para o cálculo da menor distância entre os extremos.

A Figura 3.34 mostra o fluxograma do algoritmo de determinação da saliência que será associada a minúcia do tipo bifurcação. O terceiro passo desse algoritmo, calcula as três distâncias entre os extremos utilizando-se a distância Euclidiana, calculada pela Equação (3.89).

$$D_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (3.89)$$

onde:

D_{ij} – distância Euclidiana entre os extremos de índices i e j .

(x_i, y_i) – coordenada do extremo de índice i .

(x_j, y_j) – coordenada do extremo de índice j .

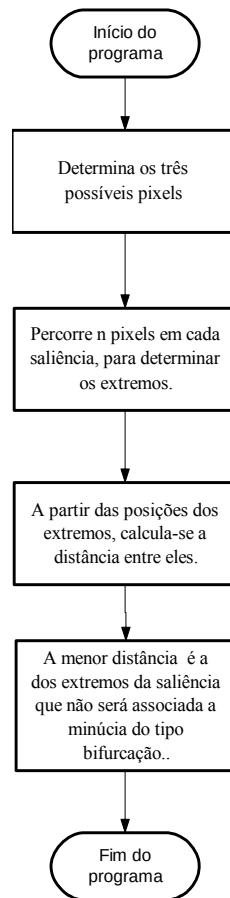


Figura 3.34 – Algoritmo de determinação da saliência que será associada a uma minúcia do tipo bifurcação.

A Figura 3.35 mostra os passos do algoritmo de determinação da saliência que será associada a minúcia do tipo bifurcação. A Figura 3.35(a) mostra o pixel classificado como minúcia de coordenada (5, 4) na cor vermelha. A direção da orientação de campo da minúcia é representada por uma linha tracejada na cor azul.

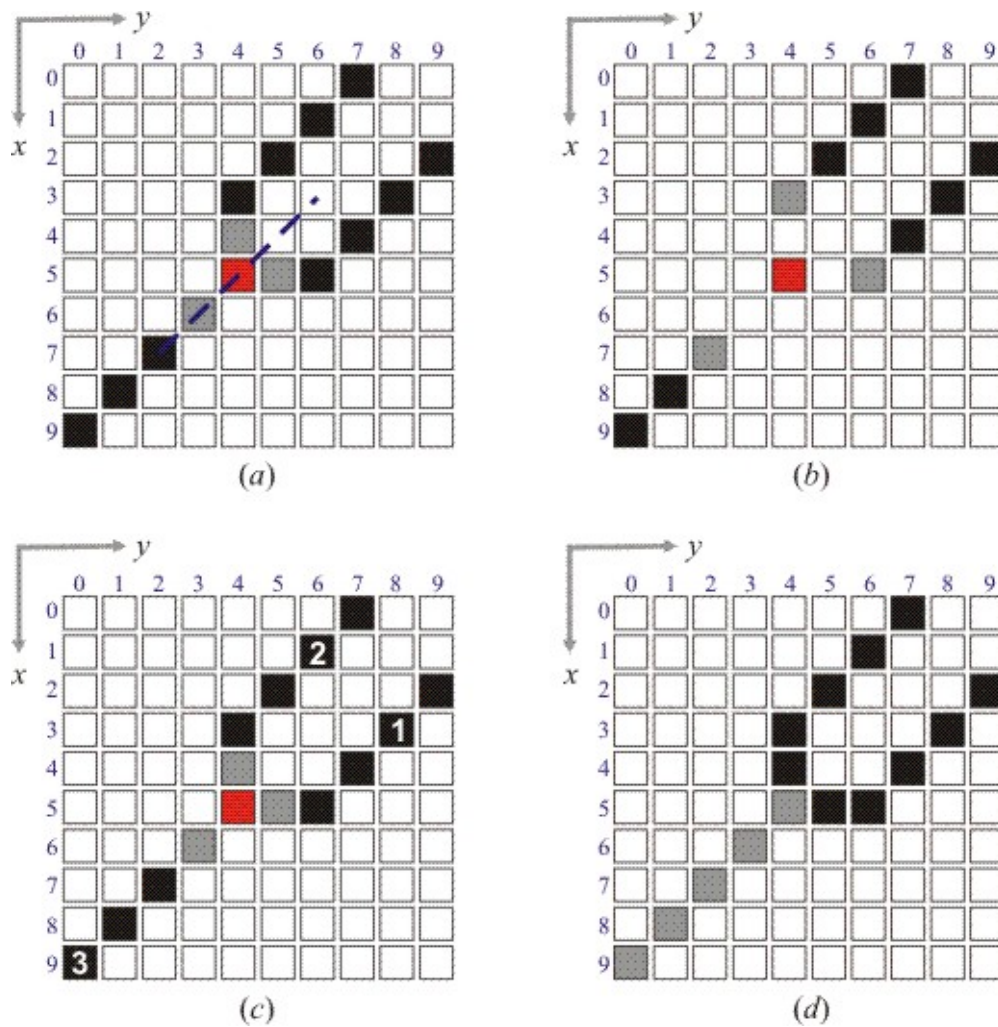


Figura 3.35 – Passos do algoritmo de determinação da saliência que será associada a uma minúcia do tipo bifurcação. (a) região 10×10 de uma minúcia; (b) três possíveis pixels; (c) extremos de cada saliência; (d) saliência escolhida na cor cinza.

O primeiro passo do algoritmo de determinação da saliência que será associada a uma minúcia do tipo bifurcação é identificar os três possíveis pixels que podem ser o segundo pixel da saliência. Na Figura 3.35(a) as coordenadas desses pixels são iguais a (4, 4), (5, 5) e (6, 3) e eles estão representados na cor cinza. Depois de determinadas as coordenadas desses três pixels, é atribuído a eles o valor igual a zero, para que o algoritmo descrito na Seção 3.9

deste capítulo (algoritmo para eliminação de falhas presentes em saliências) não escolha o pixel errado devido a ordem em que os pixels são testados.

A Figura 3.35(b) mostra os três possíveis pixels como pixels de vale (apagados no passo anterior) e a partir dos pixels de coordenadas (7, 2), (4, 3) e (5, 6), cada saliência deve ser percorrida contando n pixels (neste trabalho, $n = 10$) ou até alcançar um dos extremos de uma saliência (minúcias do tipo bifurcação, minúcias do tipo terminação ou limites da imagem). Como a imagem da Figura 3.35(c) é pequena, o valor de n é igual a 4, pois a saliência que se encontra na parte inferior da Figura 3.34(c) possui 4 pixels (além da minúcia) e quando o quarto pixel é encontrado, atingi-se o limite da imagem e o algoritmo é encerrado.

A Figura 3.35(c) mostra os extremos de cada saliência destacados pelos números 1, 2 e 3 e de coordenadas (3, 8), (1, 6) e (9, 0), respectivamente. Então, as três possíveis distâncias são calculadas, como mostra as Equações (3.90) a (3.92).

$$D_{12} = \sqrt{(3-1)^2 + (8-6)^2} = 2,8284 \quad (3.90)$$

$$D_{13} = \sqrt{(3-9)^2 + (8-0)^2} = 10 \quad (3.91)$$

$$D_{23} = \sqrt{(1-9)^2 + (6-0)^2} = 10 \quad (3.92)$$

A menor distância encontrada entre os extremos nas Equações (3.90) a (3.92) é D_{12} , ou seja, os extremos identificados por 1 e 2 pertencem as saliências que não serão associadas a minúcia.

A Figura 3.35(d) mostra os pixels da saliência (na cor cinza) que será associada a minúcia do tipo bifurcação. Então, o pixel de coordenada (6, 3) é o segundo pixel da saliência. Depois de determinado os pixels da saliência, a conversão das saliências em sinal discreto unidimensional é realizada da mesma forma que uma saliência associada a uma minúcia do tipo terminação.

Depois que todas as saliências são convertidas para sinais discretos unidimensionais é necessário a normalização desses sinais. O motivo dessa normalização, como explicado anteriormente neste capítulo, é que não exista, para uma mesma saliência com imagens de tamanhos diferentes, dois sinais discretos unidimensionais com amplitudes diferentes. Para resolver esse problema de escala todos esses sinais são divididos pela distância média entre as saliências da impressão digital que foram extraídos esses sinais.

3.12 – Cálculo da Distância Média entre as Saliências

Jain e outros [1] propuseram calcular a distância média entre as saliências para que essa distância auxiliasse no alinhamento de duas imagens de impressão digital antes do casamento de minúcias no algoritmo de reconhecimento.

O método proposto por Onnia e Tico [7] calcula a distância média entre as saliências a partir da imagem da impressão digital em níveis de cinza. A Figura 3.36 mostra a região onde será calculada a distância média entre as saliências. Nessa figura a seta de cor azul indica a direção da orientação de campo do pixel do centro do bloco, este bloco está realçado na cor amarela. O princípio desse algoritmo é obter a projeção da forma de onda de uma região limitada por uma janela orientada realçada na cor vermelha. Depois de determinada essa

projeção são localizados os máximos e a partir de suas posições é calculada a distância média entre as saliências.

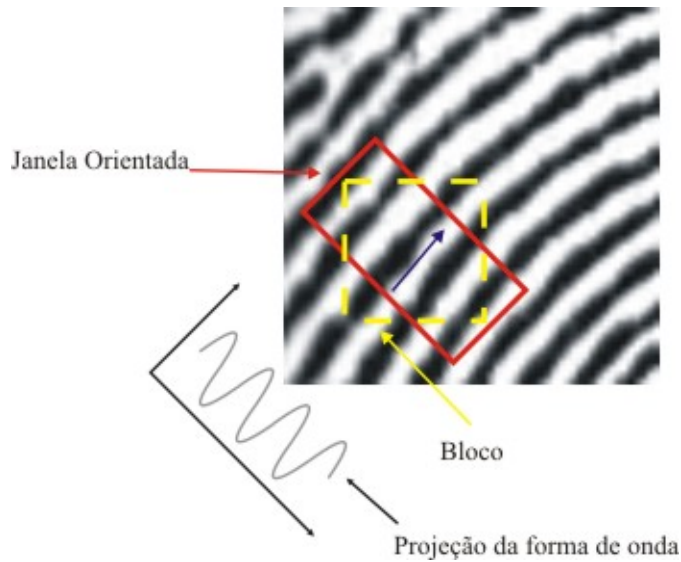


Figura 3.36 – Região onde será calculada a distância média entre as saliências.

Os passos para o cálculo da distância média entre as saliências são:

- 1º - Realiza-se a filtragem espacial da imagem da impressão digital em níveis de cinza utilizando-se uma máscara de convolução 7×7 , como mostrado na Figura 3.37;

$$\frac{1}{49} \times$$

1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1
1	1	1	1	1	1	1

Figura 3.37 – Máscara de convolução 7×7 .

2º - Divide-se a imagem em blocos de tamanho $w \times w$. Neste trabalho utiliza-se $w = 16$;

3º - Considera-se a coordenada do centro de cada bloco igual a (i, j) . Para cada bloco, calcula-se a projeção da forma de onda, $Z[0], Z[1], \dots, Z[l-1]$, das saliências e dos vales localizados dentro de uma janela orientada de dimensões $w \times l$. Neste trabalho utiliza-se $w = 16$ e $l = 32$. A inclinação dessa janela é perpendicular a orientação de campo do centro do bloco. Os valores de $Z[k]$ são obtidos utilizando-se as Equações (3.93) a (3.95);

$$Z[k] = \frac{1}{w} \sum_{d=0}^{w-1} I(u, v), \quad k = 0, 1, \dots, l-1 \quad (3.93)$$

onde:

$\theta(i, j)$ – orientação de campo do pixel de coordenada (i, j) .

$$u = i + \left(d - \frac{w}{2}\right) \cos \theta(i, j) + \left(k - \frac{l}{2}\right) \sin \theta(i, j) \quad (3.94)$$

$$v = j + \left(d - \frac{w}{2}\right) \sin \theta(i, j) + \left(\frac{l}{2} - k\right) \cos \theta(i, j) \quad (3.95)$$

4º - Encontra-se os pontos de máximos da projeção da forma de onda;

- 5º - Calcula-se a distância média entre os pontos de máximo adjacentes na projeção da forma de onda. Caso a projeção da forma de onda não possui mais de um ponto de máximo, atribui-se -1 a distância média entre as saliências desse bloco;
- 6º - Interpola-se os valores das distâncias médias de cada bloco que possui valor igual a -1 . A interpolação das distâncias é obtida utilizando-se a Equação (3.96); e
- 7º - Calcula-se a média de todos os valores de $D'(i, j)$. O resultado dessa média é a distância média entre as saliências da imagem de impressão digital.

$$D'(i, j) = \begin{cases} D(i, j), & \text{se } D(i, j) \neq -1 \\ \frac{\sum_{u=i-1}^{i+1} \sum_{v=j-1}^{j+1} D(u, v) \cdot \delta[D(u, v)]}{\sum_{u=i-1}^{i+1} \sum_{v=j-1}^{j+1} \delta[D(u, v)]}, & \text{para o restante} \end{cases} \quad (3.96)$$

3.13 – Considerações Finais deste Capítulo

Este capítulo mostrou os passos que devem ser seguidos para a extração das características de imagens de impressão digital.

Este capítulo mostrou a impressão digital e suas características, o cálculo da orientação de campo para cada pixel da imagem de impressão digital, o cálculo do nível de consistência dessa orientação, a determinação da área de interesse dessa imagem. Foram descritas as seguintes operações em relação as saliências: extração, afinamento e remoção das

falhas. Depois foram apresentadas as detecção de minúcias, a correção de falhas presentes em saliências de uma impressão digital, a determinação da área de interesse. São realizadas as seguintes operação em relação as saliências: detecção, afinamento, remoção de falhas. Depois é mostrada a detecção de minúcias, a determinação das coordenadas dos pixels de uma saliência, a conversão das saliências em sinais discretos unidimensionais e finalmente, o cálculo da distância média entre as saliências.

O próximo capítulo mostra os resultados obtidos na extração das seguintes características de imagens de impressão digital: cálculo da orientação de campo; cálculo do nível de consistência das direções encontradas no cálculo da orientação de campo, desempenho desse cálculo com e sem a utilização da propriedade da comutação, determinação da área de interesse nessas imagens; detecção, afinamento e correção das falhas presentes nas saliências; cálculo da distância média entre as saliências e a detecção das minúcias nessas saliências.

CAPÍTULO IV

RESULTADOS OBTIDOS

4.1 – Introdução

Este capítulo mostra os resultados obtidos na extração das seguintes características de imagens de impressão digital: cálculo da orientação de campo; cálculo do nível de consistência das direções encontradas no cálculo da orientação de campo, desempenho desse cálculo com e sem utilização da propriedade da comutação, determinação da área de interesse nessas imagens; detecção, afinamento e correção de falhas presentes nas saliências; cálculo da distância média entre as saliências e detecção das minúcias nessas saliências. Finalmente, são realizadas conclusões sobre esses resultados.

4.2 – Resultados Obtidos do Cálculo da Orientação de Campo

Como discutido na Seção 3.3 do Capítulo 3 deste trabalho, o cálculo da orientação de campo é o resultado do cálculo do ângulo para cada pixel da imagem $f(x, y)$. As Figuras 4.1 e 4.2 mostram o resultado do cálculo dessa orientação aplicado a duas imagens de impressão digital de tamanho 256×256 pixels com 256 níveis de cinza.

Os resultados obtidos no cálculo da orientação de campo de uma imagem utilizando o método desenvolvido por Rao [17] são visualizados nas Figuras 4.1(b) e 4.2(b), onde as orientações de campo são representadas pelos segmentos de reta realçados na cor vermelha. Para a visualização desses segmentos, a imagem é dividida em blocos de tamanhos 13×13 , onde é utilizado tanto a orientação de campo quanto a coordenada do pixel central desses blocos para desenhar o segmento de reta de cada bloco. Por esse motivo é que esses segmentos, às vezes aparecem sobre as saliências ou sobre os vales, pois o critério para desenhar cada um dos segmentos de reta é a posição do pixel central de cada bloco.

A finalidade de visualizar a orientação de campo é verificar se todas as direções estão coincidindo com as direções das saliências da imagem da impressão digital. As Figuras 4.1(b) e 4.2(b) ilustram que as orientações de campo estão coincidindo com as direções das saliências das imagens de impressões digitais.



(a)

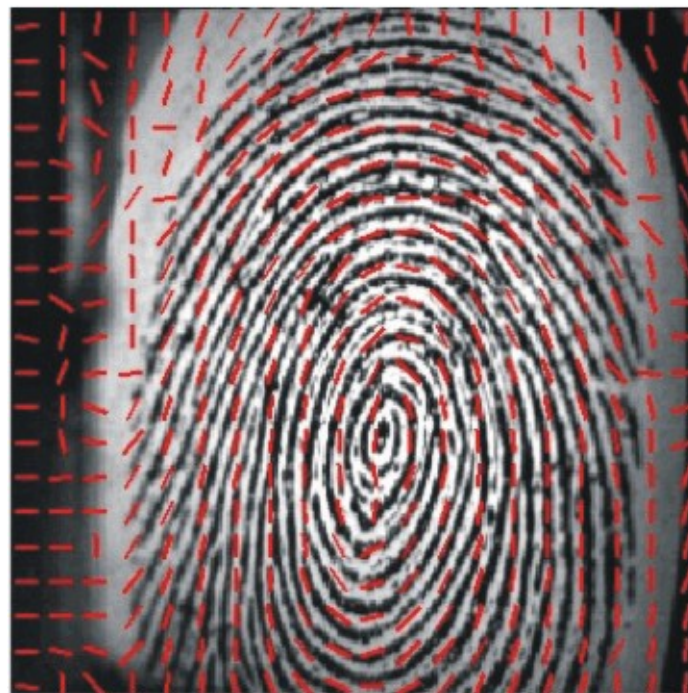


(b)

Figura 4.1 – Exemplo do cálculo da orientação de campo. (a) imagem original; (b) visualização da orientação de campo.



(a)



(b)

Figura 4.2 – Exemplo do cálculo da orientação de campo. (a) imagem original; (b) visualização da orientação de campo.