

Enriquecimento de Imagens de Impressões Digitais

Tese submetida à Faculdade de Engenharia Elétrica da
Universidade Estadual de Campinas, Departamento de
Comunicações, como requisito parcial para a obtenção
do título de Mestre em Engenharia Elétrica.

Autor

José Alexandre Nalon

Orientador

Prof. Dr. João Batista T. Yabu-uti

Campinas, 12 de dezembro de 1997.

Este exemplar corresponde a redação final da tese
defendida por JOSÉ ALEXANDRE NALON
e aprovada pela Comissão
Julgada em 12 / 12 / 97

Out: _____

N149e

33453/BC

UNICAMP
BIBLIOTECA CENTRAL

UNIDADE	BC
N.º CHAMADA:	
	Unicamp
	N.º 149e
V. E:	
TOMBO	33453
PROC.	395/98
	☐ ☐ ☒
PREÇO	R\$ 11,00
DATA	14/04/98
N.º CPD	

CM-00108590-3

FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

N149e Nalon, José Alexandre
Enriquecimento de imagens de impressões digitais. /
José Alexandre Nalon.--Campinas, SP: [s.n.], 1997.

Orientador: João Batista T. Yabu-uti
Dissertação (mestrado) - Universidade Estadual de
Campinas, Faculdade de Engenharia Elétrica e de
Computação

1. Processamento de imagens. 2. Datiloscopia. 3.
Processamento de sinais - Técnicas digitais. I. Yabu-uti,
João Batista T. II. Universidade Estadual de Campinas.
Faculdade de Engenharia Elétrica e de Computação. III.
Título.

Resumo

O objetivo deste trabalho é apresentar um algoritmo rápido e eficiente para o enriquecimento automático de imagens digitais multiníveis de impressões digitais. O algoritmo leva em conta as fortes características de orientação encontrada em cada ponto de uma impressão digital e propõe um método rápido para purificar a imagem utilizando apenas um passo de processamento, sendo realizado, portanto, com pequeno dispêndio computacional e com maior velocidade. Os resultados obtidos são analisados do ponto de vista da análise de identificação de impressões digitais, ou seja, verificando o aparecimento ou não de falsas minúcias, os detalhes utilizados pelos peritos humanos para identificar uma impressão digital.

Abstract

The objective of this work is to develop a fast and efficient algorithm to automatically enhance fingerprint images presented in gray levels. The algorithm takes into consideration the strong oriented patterns found at any point in a fingerprint image, and proposes a fast way to enhance the image in one step, thus being processed in small CPU usage and time of processing. The results obtained are analysed from the point of view of fingerprint identification, *i.e.*, by checking the occurrence of false *minutiae*, the details of a fingerprint used by experts to identify them.

Agradecimentos

Ao professor João B. T. Yabu-uti, pela orientação e pela sugestão do trabalho.

Ao professor Dalton Soares Arantes pela amizade e pela ajuda na hora necessária.

À família, pelo apoio sempre necessário.

A todos os Mestres que desde o início prepararam o caminho para que este passo
pudesse ser dado.

Ao Delegado Caviolla, pelas informações técnicas sem as quais este trabalho seria
impossível.

Ao CNPq, que financiou este trabalho.

Aos colegas da Pós-Graduação, pela camaradagem e pelas conversas em torno das
questões do Mestrado e também dos mais variados assuntos.

Ao Engenheiro Mário Niimi da Cruz, administrador da rede de computadores do
Departamento de Comunicações, por resolver todos os problemas que as máquinas
teimavam em apresentar.

A todos os amigos, longe ou perto, reais ou virtuais, sem os quais o caminho seria
muito mais difícil.

“As variáveis variam menos que as constantes.”
-Lei de Murphy aplicada à tecnologia.

Sumário

1. Introdução	1
1.1. Objetivos do Trabalho	2
1.2. Apresentação do Trabalho	2
2. Técnicas de Enriquecimento de Imagens	4
2.1. Introdução	4
2.2. Definições Matemáticas	5
2.2.1. Sistemas Lineares	5
2.2.2. Transformada de Fourier	6
2.3. Segmentação de Imagens	8
2.3.1. Introdução	8
2.3.2. <i>Thresholding</i> ou Binarização	9
<i>Thresholding</i> simples ou global	10
<i>Thresholding</i> dinâmico	11
2.3.3. Operador Gradiente	12
Gradiente Estocástico	13
2.3.4. Detecção de Bordas	15
2.4. Filtragem Espacial	15
2.4.1. Filtro Suavizador	16
2.4.2. Filtragem Passa-Alta e Passa-Banda	17
2.4.3. Filtragem Direcional	17
3. Características da Impressão Digital	19
3.1. Introdução	19
3.2. Características de uma Impressão Digital	19
3.3. Aquisição de Impressões Digitais	21
3.4. Modelagem Matemática de uma Impressão Digital	23
3.4.1. Não-estacionariedade da Impressão Digital	24
3.4.2. Segmentação da Impressão Digital	27
3.4.3. Modelagem Estatística da Impressão Digital	28
4. Enriquecimento da Impressão Digital	30

4.1. Introdução	30
4.2. Estimativa da Largura das Linhas Papilares	32
4.3. Cálculo das Máscaras de Gradiente	35
4.4. Estimativa da Região Relevante da Imagem	36
4.5. Estimativa da Orientação Local	38
4.6. Descrição do Filtro Utilizado	39
4.7. O Algoritmo Resultante	43
4.8. Limiarização da Imagem	45
4.9. Processamento em “lotes”	47
5. Resultados Obtidos	49
5.1. Implementação do Algoritmo	49
5.2. Simulações	50
5.2.1. Relação Sinal-Ruído igual a 35	51
5.2.2. Relação Sinal-Ruído igual a 0.6	53
5.2.3. Relação Sinal-Ruído igual 0.375	55
5.2.4. Relação Sinal-Ruído igual 0.3	57
5.3. Restauração em Imagens Reais	60
5.3.1. Impressão Digital em <i>loop</i>	60
5.3.2. Impressão Digital em Vórtice	62
5.3.3. Impressão Digital em Vórtice	64
5.3.4. Impressão Digital em Loop	66
5.4. Um Contra-Exemplo	69
6. Conclusão	72
6.1. Análise dos Resultados Obtidos	72
6.2. Contribuições do Trabalho	73
6.3. Sugestões para Trabalhos Futuros	74
Apêndice 1 - Máscaras de Gradiente Utilizadas	75
A1.1 - Máscara 3×3	75
A1.2 - Máscara 5×5	75
A1.3 - Máscara 7×7	76
A1.4 - Máscara 9×9	76
A1.5 - Máscara 11×11	76

A1.6 - Máscara 13×13	76
A1.7 - Máscara 15×15	76
A1.8 - Máscara 17×17	77
A1.9 - Máscara 19×19	77
A1.10 - Máscara 21×21	77
Bibliografia	78

Lista de Figuras

2.1 - Representação Esquemática de um Sistema	5
3.1 - Linhas Papilares e Vãos de uma Impressão Digital	20
3.2 - Tipos de Impressões Digitais	20-21
3.3 - Minúcias de uma Impressão Digital	21
3.4 - Período em uma Impressão Digital	24
3.5 - Representação Ideal de trechos de uma Impressão Digital	25
3.6 - Modelo Idealizado de uma Linha Papilar	26
4.1 - Imagem Exemplo de Impressão Digital a ser Enriquecida	31
4.2 - Limites da Linha Papilar	32
4.3 - Largura Local da Linha Papilar	34
4.4 - Região Relevante da Imagem Exemplo	37
4.5 - Direções da Filtragem	39
4.6 - Ação do Filtro na Linha ou no Vão	41
4.7 - Filtros Gerados para o Enriquecimento	42
4.8 - Resultado do Processamento da Imagem-Exemplo pelo Algoritmo	44
4.9 - Resultado Final do Processamento	46
5.1 - Imagem Utilizada para Testes	51
5.2 - Imagem-Teste Deteriorada com $SNR=35.0$	52
5.3 - Resultado do Processamento da Imagem 5.2	52
5.4 - Região Relevante da Imagem 5.2	53
5.5 - Imagem-Teste Deteriorada com $SNR=0.6$	54
5.6 - Resultado do Processamento da Imagem 5.5	54
5.7 - Região Relevante da Imagem 5.5	55
5.8 - Imagem-Teste Deteriorada com $SNR=0.375$	56
5.9 - Resultado do Processamento da Imagem 5.8	56
5.10 - Região Relevante da Imagem 5.8	57
5.11 - Imagem-Teste Deteriorada com $SNR=0.3$	58
5.12 - Resultado do Processamento da Imagem 5.11	58
5.13 - Região Relevante da Imagem 5.11	59
5.14 - Imagem Real de Impressão Digital em <i>Loop</i>	61

5.15 - Resultado do Processamento da Imagem 5.14	61
5.16 - Região Relevante da Imagem 5.14	62
5.17 - Imagem Real de Impressão Digital em Vórtice	63
5.18 - Resultado do Processamento da Imagem 5.17	63
5.19 - Região Relevante da Imagem 5.17	64
5.20 - Imagem Real de Impressão Digital em Vórtice	65
5.21 - Resultado do Processamento da Imagem 5.20	65
5.22 - Região Relevante da Imagem 5.20	66
5.23 - Imagem Real de Impressão Digital em Loop	67
5.24 - Resultado do Processamento da Imagem 5.23	67
5.25 - Região Relevante da Imagem 5.23	68
5.26 - Imagem Contra-Exemplo	70
5.27 - Resultado do Processamento da Imagem Contra-Exemplo	70
5.28 - Região Relevante da Imagem Contra-Exemplo	71

1. Introdução

Impressão Digital é a marca deixada pelos dedos em uma superfície qualquer. É, reconhecidamente, um dos métodos mais eficientes para a identificação que se conhece, já que as impressões digitais são únicas e imutáveis. O processo de identificação através das impressões digitais já é conhecido há bastante tempo, sendo, portanto, bastante estudado. Sua identificação tornou-se, portanto, rápida e simples, e os métodos podem ser implementados computacionalmente, se a imagem estiver em bom estado, ou seja, com baixo nível de ruído.

No entanto, a aquisição de impressões digitais pode gerar imagens ruidosas que tornam esta identificação mais difícil. Os motivos podem ser vários: se a impressão digital é adquirida na cena do crime, pode ter sido deteriorada por condições climáticas ou a superfície em que se encontra fixada não é lisa; se a imagem é adquirida por um sistema de identificação automático, o sistema de aquisição pode ser impreciso, ou a própria amostragem pode gerar algum ruído. Em qualquer um destes casos, o deslocamento dos dedos no momento da aquisição pode gerar imperfeições. Além destes, outros motivos podem também causar a deterioração da imagem.

A purificação e o enriquecimento de uma impressão digital, ainda que seja apenas o passo preliminar à identificação, é essencial para que esta seja feita com sucesso. Um bom pré-processamento da imagem evita o aparecimento de detalhes falsos que podem gerar uma identificação errônea.

Vários algoritmos foram desenvolvidos ao longo do tempo para efetuar a purificação e o enriquecimento de imagens de impressão digital. O’Gorman e Nickerson[1] apresentaram um método eficiente que, no entanto, exige alguma participação do usuário, não sendo, portanto, completamente automático. Hung[2] propôs um método baseado na uniformização das *linhas papilares*, mas seu algoritmo trabalha apenas com imagens binárias. Sherlock, Monro e Millard[3] apresentaram um método com bons resultados, mas que exige um alto dispêndio computacional, devido ao excessivo número de cálculos efetuados.

1.1 - Objetivos do trabalho

O objetivo deste trabalho foi fornecer um algoritmo eficiente que recupere e enriqueça uma imagem de uma impressão digital qualquer, de maneira completamente automática (*i.e.*, sem a participação do usuário final), de forma a retirar desta imagem o ruído que possa impedir o reconhecimento da impressão, seja ela realizada computacionalmente ou por de um perito humano.

No entanto, foi dada ênfase à qualidade visual da imagem restaurada. Devido à falta de recursos na polícia brasileira, usuária em potencial destes resultados, um sistema automático de reconhecimento de impressões digitais, embora altamente desejável, na maior parte dos casos não é possível.

O objetivo, portanto, é obter uma forma rápida e barata de recuperar uma impressão digital, facilitando o trabalho de peritos humanos que venham a fazer o reconhecimento. Adicionalmente, os resultados obtidos pelo algoritmo também serão úteis aos sistemas de reconhecimento automático, já que o aumento da relação sinal/ruído permite que a identificação seja feita com maior eficiência.

Dessa forma, pretende-se aqui obter uma imagem de alta qualidade, agradável ao olho humano, com uma alta relação sinal/ruído, de tal forma que o reconhecimento seja facilitado.

1.2 - Apresentação do trabalho

Esta tese é o resultado da procura por este algoritmo ótimo para a restauração de impressões digitais. Divide-se em seis capítulos, dos quais esta introdução é o primeiro.

O Capítulo 2 apresenta e descreve os conceitos básicos da teoria de enriquecimento de imagens, bem como as técnicas clássicas que são utilizadas no algoritmo para a restauração da impressão digital.

O Capítulo 3 descreve matematicamente e estatisticamente uma impressão digital, criando um modelo matemático e estatístico que é necessário para desenvolver os conceitos básicos do algoritmo. Em torno destes conceitos e utilizando as técnicas descritas no capítulo anterior, o algoritmo é desenvolvido.

O Capítulo 4 apresenta o algoritmo resultante passo-a-passo, ressaltando detalhes que venham a ser importantes para que a sua execução seja a mais eficiente possível.

O Capítulo 5 apresenta detalhes da implementação do algoritmo, bem como o resultado dos testes utilizados para avaliar a sua eficiência, e o resultado do processamento de imagens de impressões digitais reais.

Finalmente, o Capítulo 6 apresenta as conclusões obtidas, bem como a relevância do trabalho e sugestões para trabalhos futuros a partir de observações feitas.

2. Técnicas de Enriquecimento de Imagens

2.1 - Introdução

Uma imagem é definida como uma função de duas variáveis $f(x,y)$, sendo x e y variáveis independentes contínuas. Computacionalmente, a representação contínua de uma função é impossível, já que o computador armazena a imagem em variações discretas no espaço. Para utilizar a representação matematicamente correta, define-se uma imagem como sendo uma seqüência $f(m,n)$, sendo m e n variáveis independentes e discretas no espaço.

A seqüência $f(m,n)$ é obtida a partir da *amostragem* e *quantização* da função $f(x,y)$. A amostragem consiste em obter os valores da função em intervalos discretos na horizontal e vertical, de forma a se obter uma seqüência de valores igualmente espaçados. A quantização da imagem consiste em discretizar o valor da função em cada ponto, de forma que ela possa ser armazenada e processada computacionalmente. A amostragem e quantização fogem ao escopo deste trabalho. Maiores informações sobre estes assuntos podem ser encontradas em [4] e [5].

O enriquecimento de uma imagem é o processamento feito em uma imagem com o objetivo de acentuar uma ou mais de suas características de tal forma que essas características visadas sejam detectadas de maneira fácil, seja computacionalmente ou por um observador humano, sem obter da imagem novas informações, apenas realçando as informações desejadas.

As técnicas de enriquecimento de imagens incluem operações ponto a ponto, como a manipulação do contraste ou equalização de histograma; operações espaciais, como a redução de ruído ou filtragem; transformações, como a filtragem linear ou a filtragem homomórfica, entre outras técnicas.

As técnicas de enriquecimento de imagens são muito variadas e diferentes entre si. Além disso, muitas técnicas de enriquecimento são empíricas ou requerem interação

com o usuário de forma a obter os resultados desejados. Assim, aqui só serão discutidas as técnicas relevantes ao trabalho.

2.2 - Definições Matemáticas

2.2.1 - Sistemas Lineares

A relação entrada/saída descrita na figura 2.1 é chamada de sistema se uma seqüência utilizada como entrada gera apenas uma única seqüência de saída.

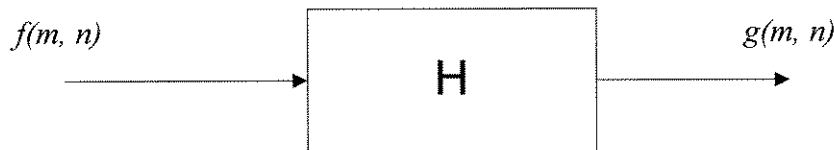


figura 2.1 - Representação esquemática de um sistema.

Um sistema H que relaciona uma entrada $f(m,n)$ com uma saída $g(m,n)$ é representado conforme a equação

$$g(m,n) = H[f(m,n)]. \quad (2.1)$$

Um sistema é linear se a saída processada de um conjunto de sinais de entrada pode ser descrita como a combinação linear do resultado do processamento de cada entrada individualmente, conforme a equação

$$H[a \cdot f_1(m,n) + b \cdot f_2(m,n)] = a \cdot g_1(m,n) + b \cdot g_2(m,n), \quad (2.2)$$

onde $g_1(m,n) = H[f_1(m,n)]$ e $g_2(m,n) = H[f_2(m,n)]$.

O sistema é invariante ao deslocamento se a aplicação de um deslocamento na seqüência de entrada causa um deslocamento na seqüência de saída, conforme mostra a equação

$$H[f(m+i, n+j)] = g(m+i, n+j), \quad i \text{ e } j \text{ inteiros} \quad (2.3)$$

A saída de um sistema linear invariante ao deslocamento é expresso pela equação

$$g(m, n) = H[f(m, n)] = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} f(i, j) \cdot h(m-i, n-j) = f(m, n) * h(m, n), \quad (2.4)$$

onde $h(m, n) = H[\delta(m, n)]$ é chamada resposta impulsiva do sistema, isto é, o resultado obtido do sistema quando a entrada é uma função impulso unitário. A operação descrita na equação 2.4 é normalmente chamada *convolução*, e é representada por um asterisco (*).

Um sistema é estável se, para cada entrada limitada, obtemos uma saída limitada. Em um sistema linear invariante ao deslocamento, essa restrição é facilmente satisfeita se a resposta impulsiva do sistema for completamente somável, como mostra a equação

$$\sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} |h(m, n)| < \infty. \quad (2.5)$$

2.2.2 - Transformada de Fourier

Qualquer sistema estável pode ser representado a partir da soma de um número infinito de exponenciais complexas de infinitas frequências, de tal forma que o sinal é representado através de suas componentes em frequência. A relação, chamada Transformada de Fourier, é dada pelo par de equações

$$f(m, n) = \frac{1}{4 \cdot \pi^2} \int_{\omega_m=-\pi}^{\pi} \int_{\omega_n=-\pi}^{\pi} F(\omega_m, \omega_n) \cdot e^{j\omega_m m} \cdot e^{j\omega_n n} \cdot d\omega_m d\omega_n \quad \text{e} \quad (2.6a)$$

$$F(\omega_m, \omega_n) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f(m, n) \cdot e^{-j\omega_m m} \cdot e^{-j\omega_n n}. \quad (2.6b)$$

onde $F(\omega_m, \omega_n)$ é a amplitude da exponencial complexa do par ordenado de frequências espaciais (ω_m, ω_n) .

Aplicando a Transformada de Fourier à relação de um sistema linear, obtemos

$$G(\omega_m, \omega_n) = F(\omega_m, \omega_n) \cdot H(\omega_m, \omega_n), \quad (2.7)$$

onde $H(\omega_m, \omega_n)$ é a Transformada de Fourier da resposta impulsiva do filtro e é chamada Função de Transferência. Dessa forma, a resposta do filtro fica completamente especificada através das suas componentes em frequência.

É útil definir

$$F(\omega_m, \omega_n) = |F(\omega_m, \omega_n)| \cdot e^{j\theta_F(\omega_m, \omega_n)}, \quad (2.8)$$

onde $|F(\omega_m, \omega_n)|$ é o módulo da Transformada de Fourier do sinal $f(m, n)$ e $\theta_F(\omega_m, \omega_n)$ é a sua fase. Se o sinal $f(m, n)$ é uma imagem, a sua característica de fase contém os detalhes finos da imagem, e pode ser considerada mais relevante, caso o processamento seja dependente deste tipo de detalhes.

Duas propriedades da Transformada de Fourier são de grande valia neste trabalho. A primeira diz que as características contínuas em uma determinada direção (por exemplo, uma linha ou uma borda) geram componentes em frequência na sua direção ortogonal (e.g., uma linha horizontal gera componentes verticais de frequência) [6].

A segunda propriedade diz que aplicar uma rotação em uma função bidimensional equivale a aplicar uma rotação do mesmo ângulo na sua representação em frequência, desde que a rotação seja feita por uma função analítica nas duas variáveis. Em outras palavras, rotacionar uma função bidimensional gera uma rotação na sua Transformada de Fourier de um ângulo equivalente [4][5]. Utilizando coordenadas polares, se fizermos

$$\begin{aligned}
m &= r \cdot \cos \theta, \\
n &= r \cdot \sin \theta, \\
\omega_m &= \rho \cdot \cos \phi, \\
\omega_n &= \rho \cdot \sin \phi,
\end{aligned}$$

a propriedade de rotação pode ser facilmente provada. A substituição no par de equações da transformada de Fourier resulta na equação

$$f(r, \theta + \theta_0) \Leftrightarrow F(\rho, \phi + \theta_0), \quad (2.9)$$

onde o símbolo $\Leftrightarrow$ representa a relação da Transformada de Fourier.

2.3 - Segmentação de Imagens

2.3.1 - Introdução

Segmentação de imagens é a decomposição de uma imagem em um ou mais objetos de interesse. Também pode ser definida como a divisão da imagem em uma ou mais regiões de interesse segundo algum critério.

Sendo quase sempre a primeira etapa do processamento de análise de uma imagem, a eficiência da segmentação pode determinar se o processamento será bem-sucedido ou não. A necessidade do processamento ditará a profundidade da segmentação.

A segmentação de imagens monocromáticas, ou com níveis de cinza, geralmente se baseia em duas características: descontinuidade e similaridade. Algoritmos baseados em detecção de bordas caem na primeira categoria, algoritmos baseados em *threshold* (binarização) caem na segunda categoria.

Em uma imagem pode haver uma ou várias regiões de interesse. No primeiro caso, a região de interesse primário é chamada objeto e a região de interesse secundário é

chamado fundo. Não faz parte do processo de segmentação, no entanto, definir o significado de cada imagem, apenas o processo de partição.

2.3.2 - *Thresholding* ou Binarização

Thresholding ou binarização consiste em segmentar uma imagem em objeto e fundo baseando-se no nível de cinza que aquela imagem apresenta em cada ponto. A suposição básica é que o objeto ou os objetos estejam com um nível de cinza bastante destacado do fundo da imagem. Dessa forma, é fácil computacionalmente classificar uma região como objeto ou fundo.

Suponha que em uma imagem, cada ponto $f(m,n)$ seja o nível de cinza da imagem. Pode-se escolher um valor T , chamado o *threshold* da imagem, que permita separar o objeto do fundo. O valor T é função de diversas variáveis:

$$T = T[m, n, f(m, n), p(m, n)], \quad (2.10)$$

onde m e n são as coordenadas do ponto tratado, $f(m,n)$ é o nível de cinza da imagem naquele ponto e $p(m,n)$ denota alguma propriedade do ponto ou de sua vizinhança. Uma imagem binarizada é definida como:

$$g(m, n) = \begin{cases} 1, & \text{se } f(m, n) \geq T \\ 0, & \text{se } f(m, n) < T \end{cases} \quad (2.11)$$

o que separa o objeto do fundo da imagem, se os pontos pertencentes ao objeto estão acima do valor T . Naturalmente, existem diversas maneiras de se escolher o valor do *threshold*, conforme será analisado a seguir. Entre as técnicas, as mais comuns, baseiam-se na análise global ou local do histograma da imagem. Técnicas mais complexas levam em conta detalhes matemáticos da imagem ou modelagens da visão humana [7], mas a análise destas técnicas não é necessária pois as técnicas mais simples são suficiente para este trabalho.

Thresholding Simples ou Global

É o tipo mais simples de binarização de uma imagem. Neste tipo de processamento, escolhe-se um valor adequado para T e aplica-se a limiarização conforme definida na equação 2.11. Normalmente, o valor do *threshold* é escolhido manualmente, após algumas tentativas, e escolhendo-se o valor que obtiver o melhor resultado.

Outra forma de se obter o valor ótimo para T é através do histograma da imagem, e essa escolha pode ser feita tanto manualmente como matematicamente. No primeiro caso, o observador analisa o histograma e escolhe o valor mais apropriado que separe o objeto do fundo da imagem.

No segundo caso, é feita a suposição de que o objeto e o fundo estão concentrados em torno de dois níveis de cinza respectivamente. A função densidade de probabilidade da imagem será dada pela equação 2.12 [4].

$$p(z) = P_o p_o(z) + P_f p_f(z), \quad (2.12)$$

onde P_o é a probabilidade do ponto (x,y) pertencer ao objeto, P_f a probabilidade deste ponto pertencer ao fundo e $p_o(z)$ e $p_f(z)$ as respectivas funções densidade de probabilidade dos *pixels* no ponto. Se considerarmos que μ_o , a média da distribuição $p_o(z)$, é o valor médio dos pontos em torno do objeto (podendo ser determinado a partir da análise do histograma da imagem), e μ_f , a média da distribuição $p_f(z)$, é o valor médio dos pontos pertencentes ao fundo, pode-se provar que o *threshold* ótimo é dado pela equação

$$T = \frac{\mu_o + \mu_f}{2} + \frac{1}{\mu_o - \mu_f} \ln \left(\frac{P_o}{P_f} \right). \quad (2.13)$$

Este valor causa o menor erro quadrático médio na limiarização da imagem. Adicionalmente, se $P_o=P_f$, o *threshold* é a média dos valores médios das distribuições. Existem outras técnicas [7] para o cálculo do limiar, mas estas técnicas fogem do escopo deste trabalho.

O *threshold* simples é eficiente em vários casos, mas tem uma grande sensibilidade ao ruído. O ruído aditivo que faça um *pixel* atravessar o limiar escolhido fará com que a decisão sobre aquele *pixel* esteja errada, gerando uma granulação indesejada na imagem binarizada.

Threshold Dinâmico

Na maior parte das aplicações, a escolha de um *threshold* simples não é suficiente para produzir bons resultados, devido às variações de baixa frequência na imagem. Uma solução é o uso do *threshold* dinâmico [4][5][6][7], que consiste na escolha de um *threshold* específico para cada ponto (m,n) da imagem com base nas estatísticas fornecidas pela sua vizinhança.

Uma forma simples de implementar o *threshold* dinâmico é estabelecer limites numéricos diferentes para o objeto e o fundo, gerando um intervalo de incerteza. *Pixels* nesse intervalo seriam considerados objeto ou fundo se a maioria de seus vizinhos fosse objeto ou fundo, respectivamente.

Outra forma é utilizar as estatísticas dos vizinhos, como a média aritmética e a variância dos seus vizinhos para calcular um *threshold* local ótimo. Se $\mu(m,n)$ é a média dos vizinhos de $f(m,n)$, e $\sigma(m,n)$ é o seu desvio-padrão, o *threshold* local é dado pela equação

$$T = k \cdot \sigma(m,n) + \mu(m,n) . \quad (2.14)$$

onde k é um valor arbitrário que indica a relevância, para a limiarização, da variância dos vizinhos do ponto.

Outras técnicas existem para realizar o *threshold* global [7], como a escolha da mediana dos vizinhos do ponto estudado.

2.3.3 - Operador Gradiente

O operador gradiente é definido como a derivada de uma função bidimensional avaliada na direção de maior variação da função no ponto estudado. Assim, é fácil ver que o operador gradiente é extremamente útil na detecção de bordas ou na avaliação da direção de propagação das bordas em uma imagem.

O gradiente de uma função bidimensional $f(x,y)$ é dado pela equação

$$\vec{\nabla}f(x,y) = \left(\frac{\partial f(x,y)}{\partial x}, \frac{\partial f(x,y)}{\partial y} \right). \quad (2.15)$$

No entanto, a expressão acima refere-se ao gradiente para o caso de uma função contínua. Pode-se fazer uma estimativa para aproximar o gradiente no caso discreto. Nesse caso, o valor de cada uma das derivadas parciais da equação 2.15 seria estimado calculando-se a convolução de uma *máscara de gradiente*, como as máscaras mostradas na tabela 2.1.

O gradiente é estimado utilizando-se uma máscara $\mathbf{G}_m$ para estimar a derivada parcial horizontal da equação 2.15 e uma máscara $\mathbf{G}_n$ para estimar a derivada vertical, conforme a equação 2.16. Normalmente, a máscara $\mathbf{G}_n$ é obtida da transposta da matriz que compõe a máscara $\mathbf{G}_m$. O gradiente de Roberts é um caso onde isso não acontece.

$$G_m(m,n) = f(m,n) * \mathbf{G}_m \quad (2.16a)$$

$$G_n(m,n) = f(m,n) * \mathbf{G}_n \quad (2.16b)$$

	G_m	G_n
Roberts	$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Prewitt	$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$
Sobel	$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$
Isotrópico	$\begin{bmatrix} -1 & 0 & 1 \\ -\sqrt{2} & 0 & \sqrt{2} \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & -\sqrt{2} & -1 \\ 0 & 0 & 0 \\ 1 & \sqrt{2} & 1 \end{bmatrix}$

Tabela 2.1 - Operadores Gradiente mais comuns [5].

O módulo e a direção do gradiente podem portanto serem calculados respectivamente utilizando-se as equações

$$|G(m,n)| = \sqrt{G_m^2(m,n) + G_n^2(m,n)} \quad e \quad (2.17a)$$

$$\theta_G(m,n) = \tan^{-1} \left(\frac{G_n(m,n)}{G_m(m,n)} \right) \quad (2.17b)$$

Gradiente Estocástico

Os operadores gradiente não costumam ter uma boa performance na presença de ruído. Pode-se solucionar essa questão com vários procedimentos, como tirar a média ou fazer uma filtragem passa-baixas antes de aplicar o operador. Uma forma mais eficiente é projetar um operador gradiente que trabalhe nas condições necessárias.

O operador gradiente pode ser formulado como a combinação linear de um preditor *forward* e um preditor *backward* ótimos [5], baseados nas estimativas de ruído e autocorrelação da imagem.

$$G(m, n) \stackrel{\Delta}{=} \hat{u}_f(m, n-1) - \hat{u}_b(m, n+1) \quad (\text{eq. 2.18})$$

onde $\hat{u}_f(m, n)$ é o preditor *forward* e $\hat{u}_b(m, n)$ é o preditor *backward*.

Os preditores podem ser calculados através de qualquer método de projeto de filtros lineares. Se considerarmos que

$$\hat{u}_f(m, n) = \sum_{(i,j) \in W} a(m, n) f(m-i, n-j), \quad (2.19)$$

onde W é uma janela semi-causal à esquerda, os coeficientes $a(m, n)$ podem ser calculados solucionando-se a equação

$$\left[\frac{\sigma_\eta^2}{\sigma^2} \delta(m, n) + r_0(m, n) \right] * a(m, n) = r_0(m, n) \quad (m, n) \in W, \quad (2.20)$$

onde σ^2/σ_η^2 é a relação sinal-ruído estimada da imagem, e $r_0(m, n)$ é a função de autocorrelação da imagem. Em forma matricial, esta equação se reduz a

$$\left[\frac{\sigma_\eta^2}{\sigma^2} \mathbf{I} + \mathbf{R} \right] \mathbf{a} = \mathbf{r}_{ff}, \quad (2.21)$$

onde $\mathbf{I}$ é a matriz identidade, $\mathbf{R}$ é a matriz de auto-correlação, $\mathbf{a}$ é o vetor dos coeficientes e $\mathbf{r}_{ff}$ é o vetor de auto-correlação.

A estimativa $\hat{u}_b(m, n)$ pode ser calculada da mesma maneira, exceto que W é uma janela semi-causal à direita. Se a função de autocorrelação $r_0(m, n)$ for simétrica, $\hat{u}_b(m, n)$ será a reflexão de $\hat{u}_f(m, n)$.

2.3.4 - Detecção de Bordas

Bordas são definidas como os limites entre os vários segmentos de uma imagem. Nada mais natural que a distinção entre o objeto e o fundo da imagem possa ser feita através da detecção das bordas do objeto. Em imagens binárias, a detecção é extremamente simples, no entanto, em imagens com vários níveis de cinza a detecção pode tornar-se um pouco mais complicada.

Existem várias formas de realizar a detecção de bordas em uma imagem. A mais simples e uma das mais eficientes é a utilização do operador gradiente.

O operador gradiente, conforme definido na equação 2.15, é a derivada de uma função de duas variáveis na direção em que há a maior variação. Portanto, o módulo do operador gradiente em um ponto fornecerá a variação da função naquele ponto. Se a variação for muito brusca, aquele ponto representa uma borda. Adicionalmente, o operador gradiente também é capaz de fornecer a direção em que a borda se propaga. Essa característica pode ser utilizada para uma detecção mais eficiente das bordas da imagem.

2.4 - Filtragem Espacial

A filtragem espacial consiste na aplicação de uma máscara a uma imagem de tal forma que as características de frequência sejam realçadas ou atenuadas conforme especificações desejadas no domínio da frequência.

A filtragem espacial é freqüentemente aplicada na forma de um filtro *FIR* (*Finite Impulse Response*) projetado tal que sua resposta aproxime o máximo possível da resposta em frequência desejada. O filtro FIR geralmente é implementado na forma de uma matriz e o resultado da filtragem é obtido através da convolução desta matriz com a imagem a ser filtrada, conforme a equação

$$g(m, n) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} f(i, j) \cdot h(m-i, n-j), \quad (2.22)$$

onde $h(m, n)$ é chamada *máscara de filtragem*.

A filtragem espacial é uma operação espacial de enriquecimento de imagens.

2.4.1 - Filtro Suavizador

Filtros suavizadores são utilizados tradicionalmente para a eliminação ou suavização do ruído, filtragem passa-baixas e reamostragem da imagem visando a diminuição da taxa de amostragem.

O filtro suavizador substitui cada *pixel* pela média aritmética dos *pixels* de sua vizinhança de tamanho estabelecido. Essa operação é implementada segundo a equação

$$g(m, n) = \sum_{i=0}^I \sum_{j=0}^J a(i, j) \cdot f(m-i, n-j), \quad (2.23)$$

onde $a(m, n)$ são os coeficientes do filtro suavizador e I e J são os limites da janela de filtragem. Se o filtro for implementado como uma simples média aritmética, então

$$a(m, n) = \frac{1}{I \cdot J}. \quad (2.24)$$

É simples notar como um filtro passa-baixas, ou filtro suavizador, diminui o ruído presente na imagem. Se considerarmos que o ruído tem média zero, e sendo

$$f(m, n) = u(m, n) + \eta(m, n), \quad (2.25)$$

onde $u(m, n)$ é a imagem original, e $\eta(m, n)$ é o ruído aditivo, então

$$\begin{aligned}
g(m,n) &= \sum_{i=0}^I \sum_{j=0}^J a(i,j) \cdot [u(m-i,n-j) + \eta(m-i,n-j)] \\
&= \frac{1}{I \cdot J} \sum_m \sum_n u(m,n) + \frac{1}{I \cdot J} \sum_m \sum_n \eta(m,n)
\end{aligned} \tag{2.26}$$

Como $\frac{1}{I \cdot J} \sum_m \sum_n \eta(m,n) \rightarrow 0$, se $I, J \rightarrow \infty$, então quanto maior for a janela utilizada, mais ruído será eliminado. No entanto, o termo $\sum_m \sum_n f(m,n)$ tende a eliminar as informações de alta frequência, dessa forma suavizando bordas, linhas e pontos que não são ruídos, mas fazem parte da imagem.

2.4.2 - Filtragem Passa-Alta e Passa-Banda

A filtragem passa-alta em enriquecimento de imagens também pode ser implementada como uma máscara, ou um filtro *FIR*. Os principais objetivos da filtragem passa-alta são a extração e a detecção de bordas e linhas, e também aumentar a definição das imagens.

A filtragem passa-banda também visa acentuar as bordas e outras características de alta frequência na presença de ruído.

2.4.3 - Filtragem Direcional

É conhecido [13] que o ruído de uma imagem filtrado na direção de propagação das bordas auxilia na visualização da imagem, enquanto o ruído filtrado na direção perpendicular a dificulta.

Pode-se, então, definir um filtro $a(i,j;m,n)$ onde os coeficientes $a(i,j)$ são dependentes da orientação da borda na região de (m,n) . Esse filtro seria altamente não-

linear e não-estacionário, pois os coeficientes $a(i,j)$ variam com o deslocamento no espaço.

A direção de propagação das bordas pode ser determinada de diversas maneiras, como a rotação e avaliação da variação do contraste no ponto especificado, ou a projeção da borda nos eixos m, n . O método mais eficiente, no entanto, é a utilização do operador gradiente para determinar a variação do contraste na horizontal e na vertical e a partir destas informações determinar a direção da borda.

$$\theta(x, y) = \tan^{-1} \left(\frac{\partial f(x, y)}{\partial y} / \frac{\partial f(x, y)}{\partial x} \right). \quad (2.27)$$

A equação 2.27 é válida para as funções analíticas e contínuas. Para as funções discretas, utiliza-se as estimativas descritas pelas equações 2.16 e 2.17.

De posse do ângulo de propagação da borda conforme a equação 2.28, a filtragem direcional pode ser implementada como

$$g(m, n) = \sum_{i=0}^I \sum_{j=0}^J a(i, j, \theta(m, n)) \cdot f(m - i, n - j). \quad (2.28)$$

3. Características da Impressão Digital

3.1 - Introdução.

A Impressão Digital é a marca deixada pelo toque do dedo em uma superfície. É, reconhecidamente, um dos métodos mais eficientes de identificação conhecidos. A identificação de uma impressão digital é simples e rápida se a impressão a ser identificada estiver em bom estado e o conjunto de impressões a ser pesquisado for pequeno.

A identificação através da impressão digital é possível em qualquer tempo por três fatores básicos. Primeiro, a configuração formada pelas *linhas papilares* é única, e não existem duas impressões digitais iguais. Segundo, a impressão digital não se altera desde a sua formação no quarto mês de gestação até mesmo depois da morte, enquanto for possível a sua aquisição. E, por último, mesmo a descamação natural da pele nos dedos ou feridas acidentais não alteram a configuração original das linhas papilares. Sanada a ferida, o desenho se reproduz de maneira idêntica.

Uma impressão digital apresenta aos olhos de um perito mais de uma centena de características, chamadas *minúcias*, que a distinguem de qualquer outra impressão existente. Geralmente, um perito utiliza de 15 a 20 dessas minúcias para identificar uma impressão digital.

3.2 - Características de uma Impressão Digital.

A impressão digital é formada pela marca deixada pelo toque dos dedos em uma superfície. Possui desenhos complexos formados por linhas finas que compõem a pele das pontas dos dedos. Estas linhas são chamadas linhas papilares. Entre cada uma destas linhas, existem os vãos, geralmente de largura inferior à das linhas. Em média, um adulto possui de 20 a 25 linhas papilares por centímetro em seus dedos. A figura 3.1 ilustra as linhas papilares e os vãos.

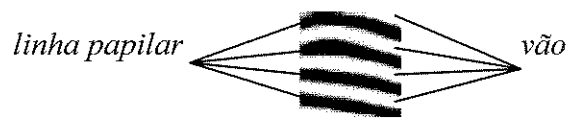


Figura 3.1 - Linhas papilares e vãos de um trecho de uma impressão digital.

A imagem é negra sobre fundo branco

As linhas e os vãos papilares perfazem um contorno complexo na impressão digital. Embora cada impressão digital seja única, esses contornos podem ser classificados em 4 tipos básicos: *loop* à direita, *loop* à esquerda, arco e vórtice. Na figura 3.2 temos a representação desses tipos básicos de contornos.

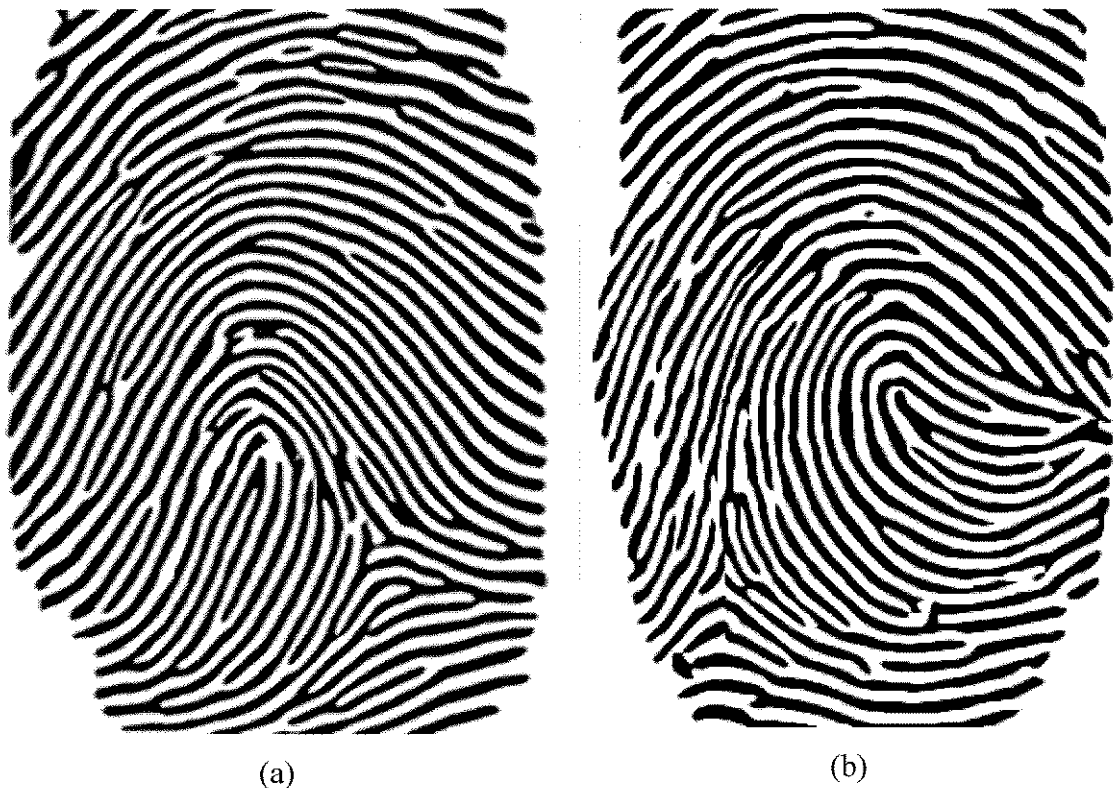


Figura 3.2 - (a) Loop à esquerda; (b) Loop à direita.



Figura 3.2(cont.) - (c) Arco e (d) Vórtice.

As impressões digitais também possuem pequenos detalhes chamados *minúcias*, que são os detalhes usados pelos peritos para a identificação. Basicamente, as minúcias são de 4 tipos: a *bifurcação*, onde a linha papilar se divide em duas; a *interrupção*, onde a linha papilar se interrompe, correspondendo a uma bifurcação do vão; a *ilha*, uma linha breve, interrompida nas duas extremidades; e o *lago*, um vão breve circundado por duas bifurcações. A figura 3.3 mostra as minúcias.



Figura 3.3 - (a) Bifurcação de uma linha papilar; (b) Interrupção; (c) Ilha e (d) Lago. As linhas estão em negro, e os vãos em branco.

3.3 - Aquisição de Impressões Digitais.

Uma impressão digital fixa-se em uma superfície de dois modos possíveis. Em materiais gelatinosos, ou moldáveis de alguma maneira, como algumas superfícies plásticas, a pressão do dedo ou da palma da mão sobre a superfície molda as linhas papilares deixando a impressão digital em relevo no material. Estas são as chamadas impressões digitais moldadas. Outra forma é através da liberação de resíduos

presentes nas pontas dos dedos ou palma da mão que se fixam na superfície tocada. No caso do resíduo ser facilmente detectável, como tinta ou pó, a impressão é dita impressão visível. No entanto, se o resíduo liberado for indetectável a uma primeira inspeção visual, como o suor proveniente da pele, a impressão é chamada impressão latente.

A impressão moldada é facilmente obtida iluminando-se lateralmente a superfície que a contém, sendo esta técnica chamada revelação por luz oblíqua. Com essa iluminação, os vãos formados no material pela pressão das linhas papilares contra ele formarão pequenas sombras que destacarão o formato da impressão digital. Uma fotografia da impressão digital pode ser tirada para análise.

Uma impressão visível não necessita de técnicas complexas para sua revelação. Se a impressão estiver em uma superfície transparente, basta colocar atrás desta superfície um fundo que faça contraste com a impressão para que ela possa ser fotografada.

No caso de impressões latentes, é necessário aplicar um revelador químico, um pó muito fino que adere ao resíduo liberado pela pele no momento do toque, fixando-se principalmente nas marcas deixadas pelas linhas papilares. Se a superfície onde a impressão está fixada for escura, aplica-se um pó branco, geralmente carbonato de chumbo, pó de alumínio ou sulfato de cálcio. Se a superfície em questão for clara, utiliza-se um pó escuro, como grafite.

Caso a superfície aonde a impressão está fixada seja colante ou gordurosa, o revelador químico aderirá a toda a superfície, tornando a técnica inútil. Neste caso, utiliza-se a iluminação oblíqua.

A impressão revelada pode portanto ser fotografada ou transferida para uma placa gelatinosa especialmente produzida para o intento. No caso da fotografia, é necessário que haja contraste suficiente entre a impressão e a superfície. Esse contraste pode ser aumentado com o uso de filtros especiais. As placas gelatinosas são pressionadas contra a impressão revelada através de pó químico, que aderirá à superfície gelatinosa. Uma fita de proteção é colocada sobre a placa para proteger a impressão obtida.

3.4 - Modelagem Matemática de uma Impressão Digital.

Para que seja possível o processamento da imagem de uma impressão digital, é necessário criar um modelo matemático eficiente sobre o qual seja possível trabalhar.

A imagem de uma impressão digital é tratada como um processo aleatório discreto em duas dimensões, $f(m,n)$, $0 \leq m < M$, $0 \leq n < N$. Logicamente, não é possível estabelecer uma expressão simples para a maioria das imagens, e não é diferente com a imagem de uma impressão digital. Portanto, é necessário criar um modelo estatístico que leve em conta certas regularidades da imagem, a serem tratadas mais adiante.

Também será levado em conta que $0 \leq f(m,n) \leq 1$, onde cada valor corresponde a um tom de cinza, sendo 0 o correspondente ao preto e 1 o correspondente ao branco. Na prática, trabalha-se com 256 níveis de cinza.

É assumido que a imagem corresponde a uma impressão escura sobre um fundo claro, ou seja, como se as impressões fossem marcadas com tinta negra em papel branco. Dessa maneira, as linhas papilares correspondem à parte escura, e os vãos à parte clara. Apesar disso ser uma simplificação, não existe perda de generalidade no tratamento matemático a ser dado.

Por fim, algumas simplificações são feitas para facilitar o modelamento. Nenhuma destas simplificações, no entanto, causa a perda da generalidade. O modelo desenvolvido é aplicável a qualquer impressão digital, obtida sob qualquer método.

3.4.1 - Não-estacionariedade da Impressão Digital

Uma das simplificações é que todas as linhas papilares têm exatamente a mesma largura, o que, embora não seja exato, facilita o desenvolvimento do método.

Por fim, considera-se como sendo um período o intervalo contendo uma linha papilar e um vão, em semelhança com uma função onda quadrada. Tanto o vão como a linha tem a mesma largura de meio período. A figura 3.4 ilustra um período em uma imagem conforme definido.

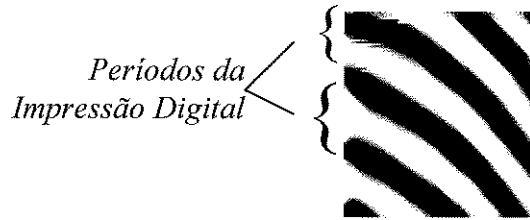


Figura 3.4 - Período em uma Impressão Digital

Uma das principais características da impressão digital é o fato de ser possível identificar com facilidade, em uma imagem nítida, a orientação local de uma linha. A cada região é possível associar uma orientação dada sob a forma de um ângulo, sendo 0° uma linha horizontal.

$$\theta = \theta(m, n) \quad (3.1)$$

É fácil mostrar que o valor médio dos pontos em cada região é constante. Admitindo-se que a densidade das linhas papilares e dos vãos são, respectivamente, d_l e d_v , sendo que, idealmente, os pontos em uma linha tem valor 0 (ou seja, negro) e os pontos em um vão tem valor 1 (ou seja, branco), então

$$\bar{f}(m, n) = \frac{1}{I \cdot J} \sum_{i=0}^I \sum_{j=0}^J f(m+i, n+j) = 0 \cdot d_l + 1 \cdot d_v, \quad (3.2)$$

onde $I \times J$ é o tamanho da região analisada.

Se aplicarmos um deslocamento no processo aleatório $f(m, n)$, obteremos

$$\begin{aligned}\bar{f}(m+p, n+q) &= \frac{1}{I \cdot J} \sum_{i=0}^I \sum_{j=0}^J f(m+p+i, n+q+j) = \\ &= 0 \cdot d_l + 1 \cdot d_v = \bar{f}(m, n)\end{aligned}\quad (3.3)$$

Portanto, não há variação da média com o deslocamento. Um raciocínio semelhante pode ser feito com o cálculo da variância para o mesmo modelamento, e a conclusão é a mesma. No entanto, utilizando-se o mesmo raciocínio para o cálculo da autocorrelação, pode-se provar que isso não se repete.

Seja então θ_l a orientação local das linhas na região (m, n) e θ_2 a orientação na região $(m+p, n+q)$. Para simplificar o desenvolvimento do cálculo da autocorrelação de $f(m, n)$, sem perda da generalidade, considera-se que tanto θ_l como θ_2 são menores que 45° , e que, localmente (isto é, em uma região de tamanho menor que a janela estudada), a variação da orientação é desprezível, não influenciando consideravelmente no cálculo. Na figura 3.5, temos a representação das funções.

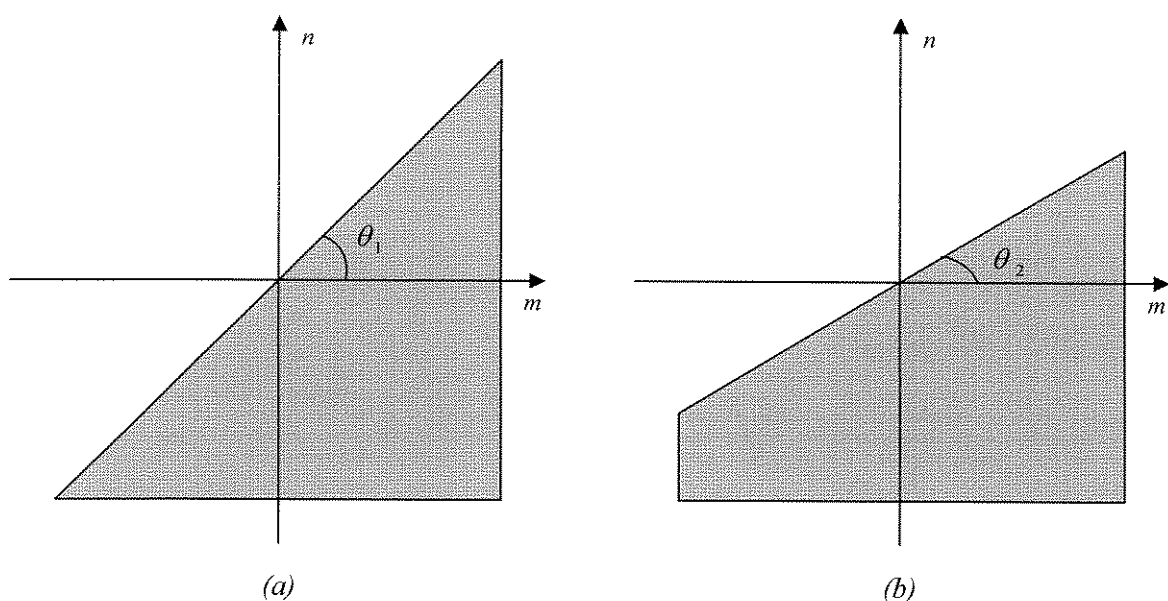


Figura 3.5 - Representação ideal de dois trechos de uma mesma impressão digital

A linha papilar pode ser definida simplificadaamente como sendo a região entre as retas $n = m \cdot \tan^{-1} \theta - J/2$ e $n = m \cdot \tan^{-1} \theta + J/2$, $-I/2 \leq m \leq I/2$, conforme pode ser visto na figura 3.6.

Portanto, pode-se definir a seguinte expressão para representar uma linha papilar simples

$$f(m,n) = \begin{cases} 1, & m \cdot \tan^{-1} \theta - J/2 \leq n \leq m \cdot \tan^{-1} \theta + J/2, \quad -I/2 \leq m \leq I/2 \\ 0, & \text{fora do intervalo} \end{cases} \quad (3.4)$$

Calcula-se a autocorrelação de $f(m,n)$ como sendo

$$\begin{aligned} r_{ff}(k,l) &= \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f(m,n) \cdot f(m-k,n-l) = \\ &= \sum_{m=-I/2}^{I/2} \sum_{y=(m+k) \cdot \tan^{-1} \theta_1 - J/2}^{m \cdot \tan^{-1} \theta_1 + J/2} 1 = IJ + J|k| \arctan \theta_1 \end{aligned} \quad (3.5)$$

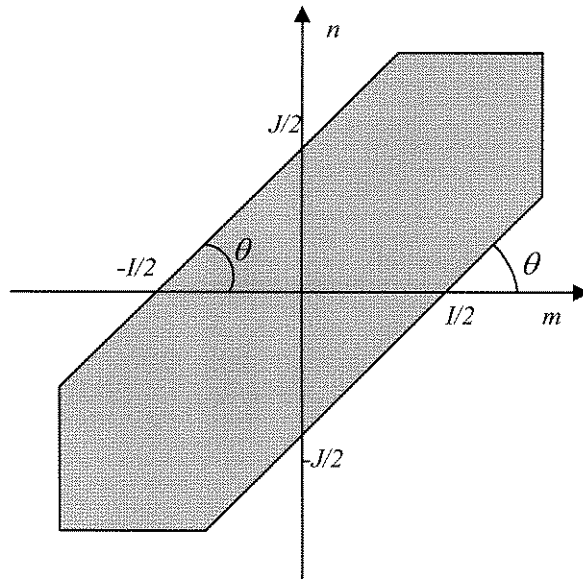


Figura 3.6 - Modelo idealizado de uma linha papilar em uma impressão digital.

Similarmente, para $f(m+p, n+q)$, obtemos

$$r_{ff}(k, l) = IJ + J|k| \arctan \theta_2, \quad (3.6)$$

No entanto, como $\theta_1 = \theta(m, n)$ e $\theta_2 = \theta(m+p, n+q)$, a autocorrelação da imagem estudada depende da região onde se faz o estudo. Dessa forma, a imagem não pode ser considerada estacionária, pois a autocorrelação é variante ao deslocamento.

3.4.2 - Segmentação da Impressão Digital

Uma imagem não-estacionária é muito mais difícil de ser trabalhada, pois os filtros mais estudados funcionam apenas com imagens estacionárias. A solução para o problema, no entanto, é bastante simples. Basta segmentar a imagem em pequenas janelas onde as estatísticas não variem com o deslocamento. Dessa forma, teremos sub-imagens que podem ser consideradas estacionárias, onde é possível aplicar filtros lineares otimizados para as características estatísticas de cada região.

A filtragem e o filtro resultante serão não-estacionários e não-lineares, embora cada um de seus componentes o seja.

Como visto anteriormente, a principal causa da não-estacionariedade da imagem de uma impressão digital é a orientação local das linhas. No entanto, outros fatores, como a largura das linhas papilares ou dos vãos, também podem influir na não-estacionariedade da impressão digital.

Segmenta-se, então, de forma que em cada sub-imagem não haja uma variação significativa na largura das linhas bem como em sua orientação. A região, no entanto, deve ser abrangente o suficiente para conter todas as informações pertinentes a pelo menos uma linha e um vão. Se considerarmos que $W = W_l + W_v$, onde W_l é a largura média de uma linha e W_v a largura média de um vão, W será a largura média em *pixels* de um período (*i.e.* uma linha e um vão), podemos dividir a impressão digital em

várias regiões $W \times W$ contendo toda a informação local referente à impressão digital sem que haja variação da sua orientação ou da largura da linha ou do vão.

Assim sendo, cada sub-imagem dividida desta forma é por si uma imagem estacionária, podendo ser tratada de maneira simples.

3.4.3 - Modelagem Estatística da Impressão Digital

Tendo a imagem sido dividida em janelas de $W \times W$ *pixels*, onde cada sub-imagem é estacionária, cada janela pode ser filtrada linearmente.

Além de cada uma das sub-imagens que fazem parte da impressão digital, algumas sub-imagens podem conter dados irrelevantes, como por exemplo, as bordas do papel onde não há impressão digital. Estas janelas podem ser ignoradas no processo de enriquecimento, sendo substituídas pela informação de fundo da imagem. Mais tarde veremos como detectar a região de interesse. Portanto, o modelo matemático proposto aqui vale apenas para as sub-imagens significativas.

Estabelece-se, então, um modelo para cada janela contendo, em média, um período. Como o único objetivo deste modelo é obter um modelo de autocorrelação, várias simplificações são feitas. Inicialmente, é considerado que a largura da linha é a mesma largura do vão. Segundo, é considerado que a imagem é binária e perfeita, ou seja, sem recortes ou imperfeições nas bordas. Outra hipótese é que temos a imagem em preto ($f(m,n) = 0$) sobre fundo branco ($f(m,n) = 1$). Por último, a orientação não é levada em conta, pois o modelo de autocorrelação para qualquer orientação pode ser obtido através da rotação do modelo horizontal.

Segundo estas hipóteses simplificadoras, obtém-se o seguinte modelo

$$f(m,n) = \begin{cases} 0, 0 \leq m < W, 0 \leq n < W/2 \\ 1, 0 \leq m < W, W/2 \leq n < W \end{cases} \quad (3.7)$$

O modelo de autocorrelação pode então ser facilmente obtido

$$r_{ff}(k, l) = \sum_m \sum_n f(m, n) \cdot f(m + k, n + l) = (W - |k|) \left(\frac{W}{2} - |l| \right). \quad (3.8)$$

Essa relação é extremamente útil no desenvolvimento das máscaras de gradiente e dos filtros que são utilizados no enriquecimento da imagem.

4 - Enriquecimento da Impressão Digital

4.1 - Introdução

O algoritmo aqui apresentado foi desenvolvido visando aproveitar o máximo das características físicas e estatísticas descritas no capítulo 3. A principal característica a ser explorada é o fato que a impressão digital é fortemente orientada, ou seja, as linhas papilares e os vãos determinam uma orientação fácil de ser detectada em cada ponto da imagem. Realçar esse padrão orientado, portanto, conferirá uma maior nitidez à imagem, o que a torna alvo perfeito para uma filtragem direcional, como descrita na seção 2.4.4.

O algoritmo usa outros recursos descritos no capítulo 2 para otimizar a filtragem direcional para a impressão digital. A saber: detecção de bordas para estimar a largura das linhas e a região relevante da imagem, realizando esta operação através dos operadores gradiente descritos; detecção da orientação das linhas e vãos também através do uso de operadores gradiente; filtragem direcional para a restauração das linhas e vãos e a limiarização da imagem para a obtenção do resultado final com o maior nível de contraste possível. Cada um destes passos será descrito com maior profundidade ao longo deste capítulo.

Fundamentalmente, o algoritmo segue os seguintes passos:

- 1 - Estima a largura média das linhas papilares e vãos;
- 2 - Estima a região relevante da imagem;
- 3 - Estima a orientação local das linhas papilares e dos vãos;
- 4 - Realiza a filtragem direcional;
- 5 - Limiariza a imagem;

O primeiro passo do algoritmo é fundamental para todo o restante, pois determinar a largura das linhas indica qual o tamanho da janela a ser utilizada para se segmentar a imagem. Os passos 2, 3 e 4, apesar de serem seqüenciais e necessitarem serem executados nesta ordem, podem ser feitos ponto-a-ponto, pois o resultado obtido ao se

processar o ponto (m,n) não depende do resultado nos demais pontos. A limiarização, o quinto passo, pela sua própria natureza, depende dos resultados obtidos nos passos anteriores e também depende do resultado obtido em outros pontos da imagem. Deve, portanto, ser executado após a execução dos passos anteriores na imagem inteira.

O algoritmo foi implementado realizando os passos 2, 3 e 4 ponto-a-ponto, pois exige um menor tempo de processamento, menor espaço em disco e menor quantidade de memória para ser executado. No entanto, aplicar cada um destes passos integralmente à imagem e depois combinar os resultados (como se fosse um processamento “em lotes”) pode trazer alguns benefícios. Essa forma de processamento também será discutida neste capítulo.

A figura 4.1 é uma impressão digital que será utilizada para ilustrar cada um dos passos do procedimento.



Figura 4.1 - Impressão Digital a ser utilizada como exemplo.

4.2 - Estimativa da largura das linhas papilares

Seja $f(m,n)$ uma impressão digital. Em qualquer um dos pontos (m,n) da imagem, podemos traçar uma linha em cada uma das direções dos eixos ortogonais m e n , conforme mostrado na figura 4.2. A partir do ponto (m,n) percorre-se estas linhas em busca dos limites do vão ou da linha onde o ponto se encontra.

Aqui não faz muita diferença se o ponto (m,n) de onde se inicia a busca pertence a um vão ou a uma linha papilar. Invariavelmente, ao se percorrer os eixos ortogonais, os limites desta linha ou deste vão serão encontrados, não sendo necessário, portanto, uma diferenciação entre linhas ou vãos.

Parte-se do ponto (m,n) em cada uma das direções, verificando-se, a cada *pixel*, se aquele ponto é ou não o limite entre a linha e o vão pesquisado. Este limite pode ser detectado com o uso de um operador gradiente simples, como o operador de Sobel ou qualquer outro mostrado na tabela 2.1.

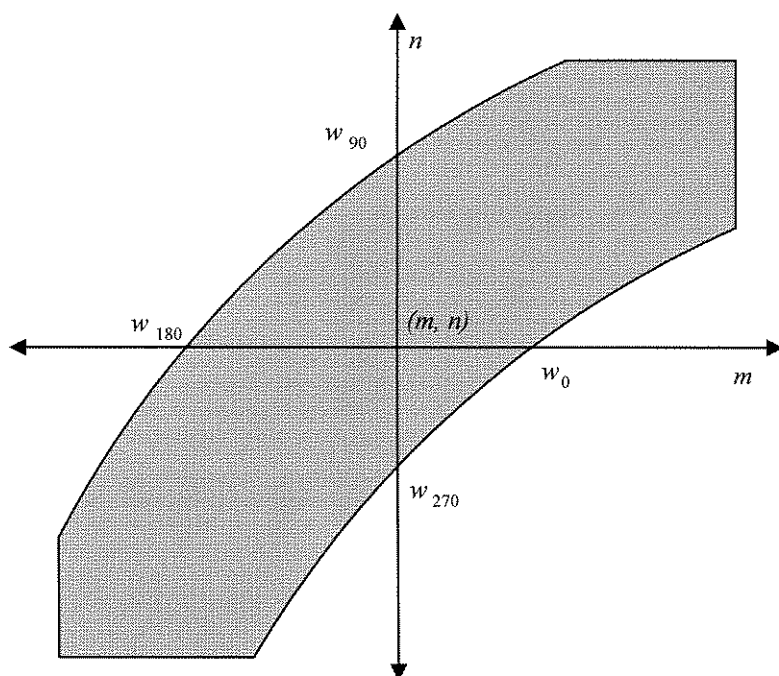


Figura 4.2 - Encontrando os limites da linha papilar.

Percorrendo-se a imagem nas quatro direções, obteremos as distâncias de que o ponto (m,n) se separa dos limites da linha ou vão onde se encontra. A essas distâncias denominamos w_0 , a distância entre o ponto (m,n) e o limite à direita; w_{90} , entre o ponto e o limite superior; w_{180} , entre o ponto e o limite à esquerda; e w_{270} , entre o ponto e o limite inferior.

Pode acontecer que o ponto (m,n) esteja localizado em uma linha perfeitamente horizontal ou vertical naquela região estudada. Estabelece-se, para estes casos, um limite máximo para a estimativa dos valores w_0 , w_{90} , w_{180} e w_{270} . Esse limite, w_{max} , é estabelecido como M , a altura em pixels da imagem, para o caso vertical, ou N , a largura em pixels da imagem, para o caso horizontal. Ao detectar um valor desta magnitude, sabe-se que se trata de uma linha na horizontal ou na vertical.

Os valores w_0 e w_{180} , e w_{90} e w_{270} são combinados para se obter respectivamente a largura e a altura da linha naquele ponto. Define-se, portanto, $w_m = w_0 + w_{180}$ como a largura local e $w_n = w_{90} + w_{270}$ como a altura local.

Para determinar a largura local w , se w_m for maior ou igual a M , trata-se de uma linha horizontal, portanto $w = w_n$. Similarmente, se w_n for maior ou igual a N , trata-se de uma linha vertical, portanto $w = w_m$. O caso mais comum, no entanto, é quando a largura e a altura não atingem os valores máximos.

Neste caso, assume-se que localmente a curvatura da linha papilar ou do vão não é significativa e pode ser desprezada, e uma relação de triângulo pode ser utilizada, como mostra a figura 4.3a. Considera-se que w_m e w_n são os catetos de um triângulo retângulo, sendo a borda da linha papilar ou do vão a sua hipotenusa. A largura local w será dada pela altura do triângulo em relação à hipotenusa, como mostra a figura 4.3b. Essa relação é dada pela equação

$$w = \frac{w_m \cdot w_n}{\sqrt{w_m^2 + w_n^2}} \quad (4.1)$$

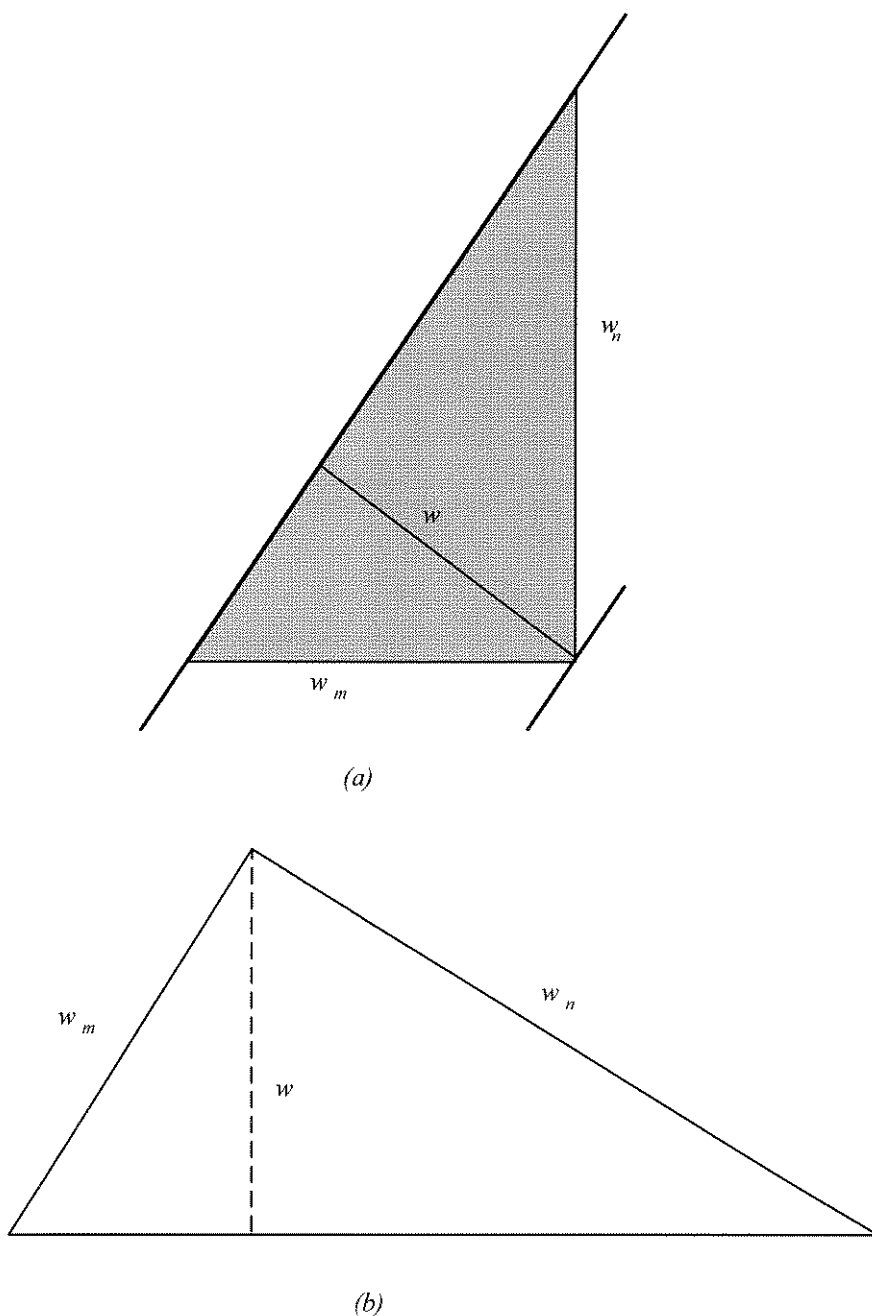


Figura 4.3 - Modo de obter a largura local da linha

Vale lembrar, no entanto, que esta estimativa é feita na imagem ruidosa. Portanto, o ruído pode levar a um erro na estimativa. Além disso, a largura da linha papilar varia ao longo da impressão digital. Logo, é necessário realizar várias estimativas ao longo da imagem para obter um valor médio w , que será usado para calcular o tamanho da janela que processará a imagem.

4.3 - Cálculo das máscaras de gradiente

Tendo definido a largura das linhas papilares, podemos definir qual a largura da janela a ser utilizada para processar a imagem. Como foi dito antes, assume-se que as linhas e os vãos têm a mesma largura. Assim, $W=2w$ parece ser a melhor escolha, já que desejamos que nossa janela abranja um período completo. No entanto, em processamento de imagens, uma janela de tamanho ímpar costuma ser uma melhor escolha, pois isso permite “centralizar” o ponto processado na máscara. Dessa forma, escolhemos $W=2w+1$.

De posse da função de autocorrelação (equação 3.5), conforme calculada no capítulo 3, basta solucionar a equação 2.19 para obtermos as máscaras de gradiente estocásticos desejadas. Relembrando a equação 2.19.

$$\left[\frac{\sigma_{\eta}^2}{\sigma^2} \delta(m,n) + r_{ff}(m,n) \right] * a(m,n) = r_{ff}(m,n) \quad (2.19)$$

É mais fácil solucionar (ou visualizar) esta equação em sua forma matricial, como na equação 2.20.

$$\left[\frac{\sigma_{\eta}^2}{\sigma^2} \mathbf{I} + \mathbf{R} \right] \mathbf{a} = \mathbf{r}_{ff} \quad (2.20)$$

Os coeficientes $a(m,n)$ a serem calculados são os coeficientes dos operadores gradiente $\mathbf{G}_m$ e $\mathbf{G}_n$. Como foi dito anteriormente, basta utilizar uma janela semi-causal, pois utilizaremos a simetria do operador gradiente para reduzir o número de cálculos. O problema reduz-se ao sistema de equações da equação

$$\sum_{k=0}^W \left[\frac{\sigma_{\eta}^2}{\sigma^2} \delta(m,k) + r_{ff}(m,k) \right] \times a(k,n) = r_{ff}(m,n). \quad (4.2)$$

Isso nos fornece um total de $w(2w+1)$ equações a serem resolvidas, de forma que se encontra os coeficientes da máscara de gradiente. A solução desse sistema de equações pode ser um tanto complicada, devido ao grande número de incógnitas, se a janela W for muito grande. No entanto, existem diversos métodos computacionais que permitem encontrar essa solução. E, como o modelo de autocorrelação é válido para qualquer impressão digital (já que é um modelo idealizado), o cálculo das máscaras de gradiente só precisam ser feitos uma vez.

A resolução do sistema de equações é demorada. Por esse motivo, é preferível realizar este cálculo apenas uma vez e armazenar o resultado em arquivos. Quando o algoritmo detecta a largura da impressão digital, apenas carrega a máscara necessária do arquivo pré-gravado.

O Apêndice 1 contém as máscaras de gradiente calculadas e utilizadas neste trabalho.

4.4 - Estimativa da região relevante da imagem

A região relevante da imagem é a região da impressão digital que contém linhas e vãos, e nessas regiões é necessário aplicar o filtro de modo a recuperar a imagem. A região da imagem que contém apenas informação de fundo pode ser substituída por um valor padrão, não havendo necessidade de processamento nesta área.

Com as máscaras de gradiente calculadas, é possível detectar a região relevante da imagem. A impressão digital é constituída de linhas e vãos de largura estimada conforme descrito na seção 4.2. O limite entre as linhas e os vãos pode ser facilmente detectado utilizando o operador gradiente.

Para determinar se um ponto pertence à região relevante da imagem, basta detectar se existe uma borda próxima a esse ponto. Essa detecção pode ser feita utilizando o

operador gradiente. No entanto, o uso de um operador gradiente simples, como os mostrados na tabela 2.1 não é o suficiente.

Conforme descrito na seção 4.2, as linhas ou os vãos de uma impressão digital têm uma largura média em *pixels*. Se a máscara de gradiente for menor que a largura das linhas, apenas os pontos da linha e do vão que estiverem nas bordas serão considerados relevantes.

É necessário utilizar uma máscara de gradiente que abranja pelo menos um período da impressão digital (*i.e.*: uma linha e um vão). Dessa forma, se nas proximidades do ponto estudado houver uma borda, ela será detectada e o ponto será considerado relevante. Se o ponto pertencer à região irrelevante, o gradiente não será detectado e o ponto não será processado.

A figura 4.4 mostra a região relevante da imagem exemplo.

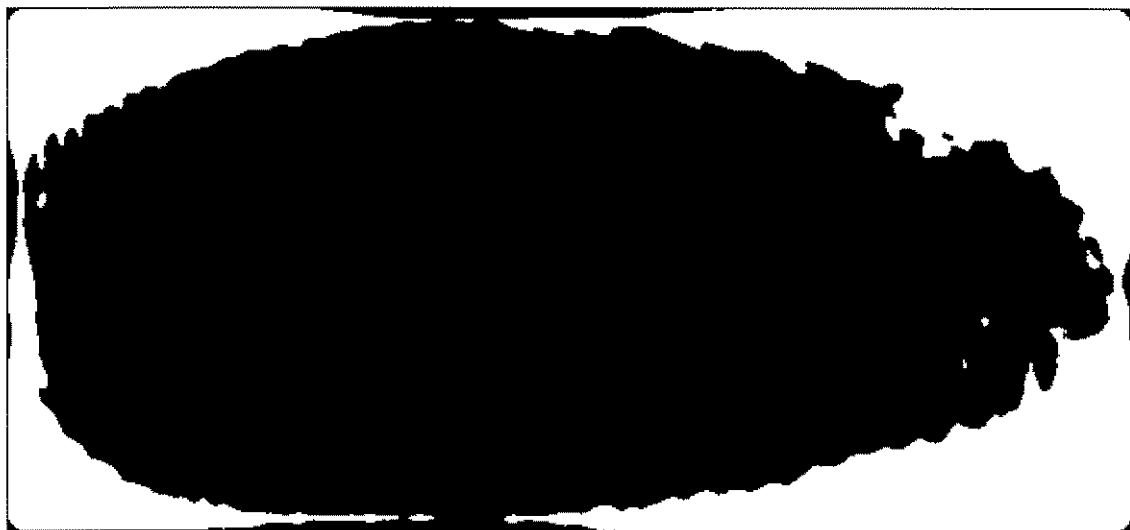


Figura 4.4 - Região relevante da imagem exemplo

4.5 - Estimativa da orientação local

Para que seja feita a filtragem direcional é necessário encontrar a orientação local das linhas. Existem várias maneiras de se estimar essa orientação, mas a mais rápida e eficiente (*i.e.*: exige menos cálculos e cálculos mais simples) é a utilização do operador gradiente.

A orientação local é definida como

$$\theta(x, y) = \tan^{-1} \left(\frac{\frac{\partial f(x, y)}{\partial y}}{\frac{\partial f(x, y)}{\partial x}} \right). \quad (4.3)$$

A equação 4.3 refere-se ao caso contínuo. No caso de uma imagem digital, basta aplicar as máscaras de gradiente $G_m(m, n)$ e $G_n(m, n)$ e calcular a orientação local como sendo

$$\theta(m, n) = \tan^{-1} \left(\frac{G_n(m, n)}{G_m(m, n)} \right). \quad (4.4)$$

Para aumentar a eficiência e a velocidade do algoritmo, é desejável trabalhar com um número reduzido de direções para a filtragem. Dessa maneira, não é necessário rotacionar os filtros a cada ponto processado (esta passagem será explicada na seção 4.6). Trabalha-se com um número limitado de direções, conforme a figura 4.5.

Como não há diferença entre a filtragem feita em uma direção e sua direção oposta (*e.g.*: 0° e 180°), consideram-se apenas as D direções escolhidas entre 0° e 180° , espaçadas igualmente. A cada uma das direções d corresponde um filtro $H_d(\cdot)$ (a ser descrito na seção 4.6) que faz a filtragem otimizada para aquela direção.

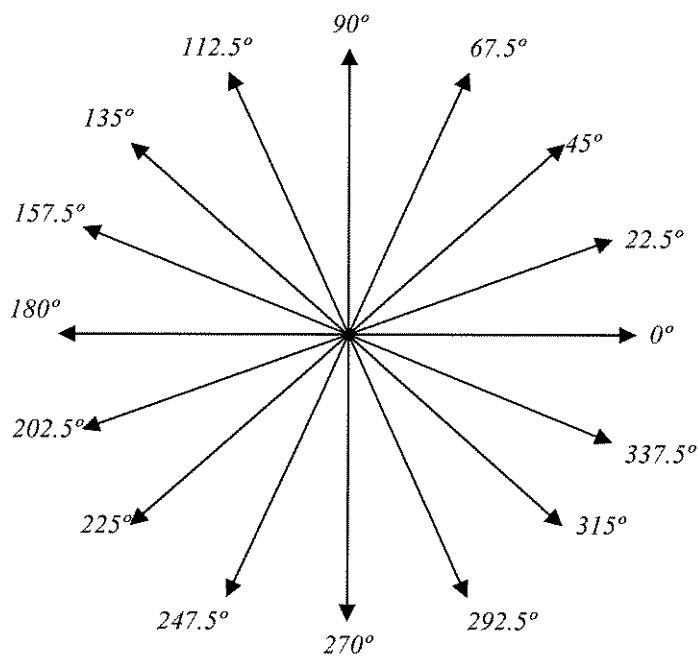


Figura 4.5 - Direções em que a filtragem é feita.

A orientação local, por fim, fica sendo dada por

$$d(m,n) = \left\lfloor \frac{\theta(m,n)}{\frac{\pi}{D}} + 0.5 \right\rfloor \times \frac{\pi}{D}, \quad (4.5)$$

onde $\lfloor d(m,n) \rfloor$ é o maior inteiro menor que $d(m,n)$, e o fator 0.5 somado é para arredondar o valor obtido para o inteiro mais próximo.

4.6 - Descrição do filtro utilizado

Como descrito na seção 3.4.1, uma imagem de uma impressão digital é altamente não-estacionária, e portanto não pode ser processada por um filtro linear simples. É necessário utilizar um filtro que se adapte às estatísticas locais do ponto processado.

Antes de descrever o filtro, é necessário saber como o filtro deve agir para enriquecer a imagem da maneira mais eficiente possível. Relembrando o

modelamento feito na seção 3.4, a impressão digital pode ser localmente definida como

$$f(m,n) = \begin{cases} 1, & m \cdot \tan^{-1} \theta - W/2 \leq n \leq m \cdot \tan^{-1} \theta + W/2, \quad -W/2 \leq m \leq W/2 \\ 0, & \text{fora do intervalo.} \end{cases} \quad (4.6)$$

Esta é a forma ideal do vão ou da linha. O ruído adicionado neste modelo tem dois efeitos. O primeiro é causar a perda da uniformidade na linha ou no vão, ou seja, causar falhas na linha que dificultam a detecção. O segundo efeito é a perda de contraste entre a linha e o vão, tornando também mais difícil a identificação das minúcias.

O que se deseja do filtro é que ele restaure o contraste entre a linha e o vão, ou, em outras palavras, ressalte a borda entre estas duas regiões e, adicionalmente, elimine as falhas na linha ou no vão causados pelo ruído.

Para eliminar as falhas, a solução é fazer uma filtragem passa-baixas, ou seja, uma suavização, mas apenas na direção da propagação das linhas. Se feita na direção perpendicular a essa propagação, o contraste entre linha e vão vai desaparecer.

Similarmente, para aumentar o contraste entre as linhas e os vãos, utiliza-se uma filtragem passa-altas. Mas esta filtragem deve ser otimizada para trabalhar na direção perpendicular à propagação das linhas, do contrário, as falhas internas às linhas e vãos seriam ressaltados.

A figura 4.6 ilustra a ação desejada do filtro.

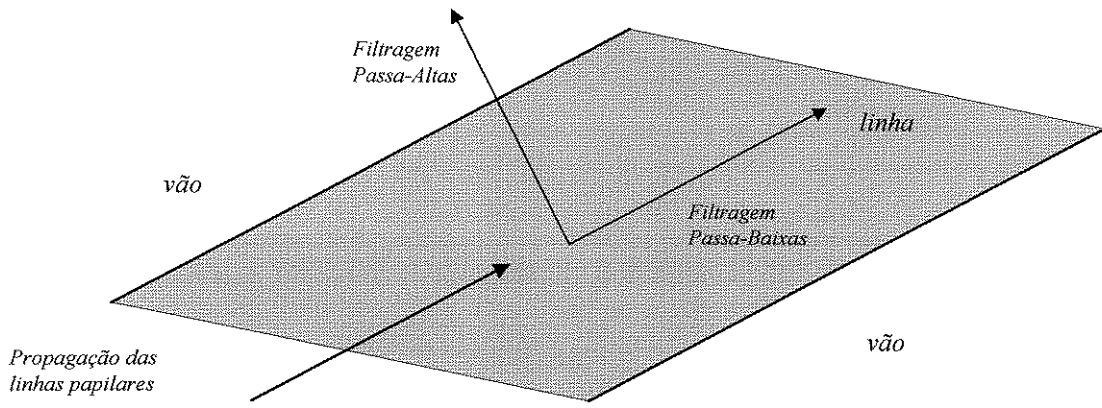


Figura 4.6 - Ação do filtro na linha ou no vão.

A definição de um filtro que siga estas especificações é simples. Inicialmente, supõe-se que a orientação da linha seja horizontal (depois pode-se extrapolar o filtro encontrado para trabalhar em qualquer direção). A suavização é feita por um filtro definido pela equação

$$H_{pb}(\omega_m, \omega_n) = H_{pb}(\omega_n), \quad (4.7)$$

onde $H_{pb}(\cdot)$ é uma expressão de um filtro passa-baixas, como o filtro de Butterworth, por exemplo. A suavização horizontal é definida para componentes verticais de frequência porque características contínuas horizontalmente na imagem (como uma borda na horizontal) se traduzem em componentes ao longo do eixo vertical das frequências.

Similarmente, a filtragem passa-altas, feita na direção ortogonal à suavização, é definida pela equação

$$H_{pa}(\omega_m, \omega_n) = H_{pa}(\omega_m). \quad (4.8)$$

Portanto, o filtro resultante será um filtro separável dado pela equação

$$H(\omega_m, \omega_n) = H_{pa}(\omega_m) \cdot H_{pb}(\omega_n). \quad (4.9)$$

A figura 4.7a mostra um filtro ideal construído desta forma. A figura 4.7b mostra o filtro construído a partir das expressões do filtro de Butterworth de 2ª ordem.

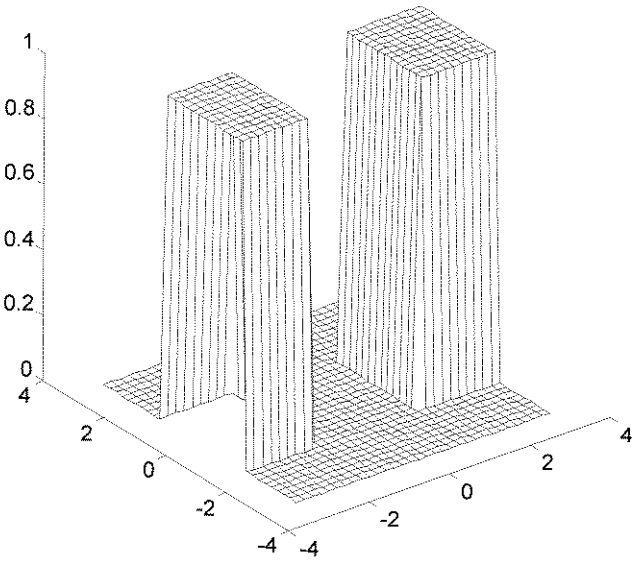


Figura 4.7a - Filtro Ideal gerado pelo método descrito.

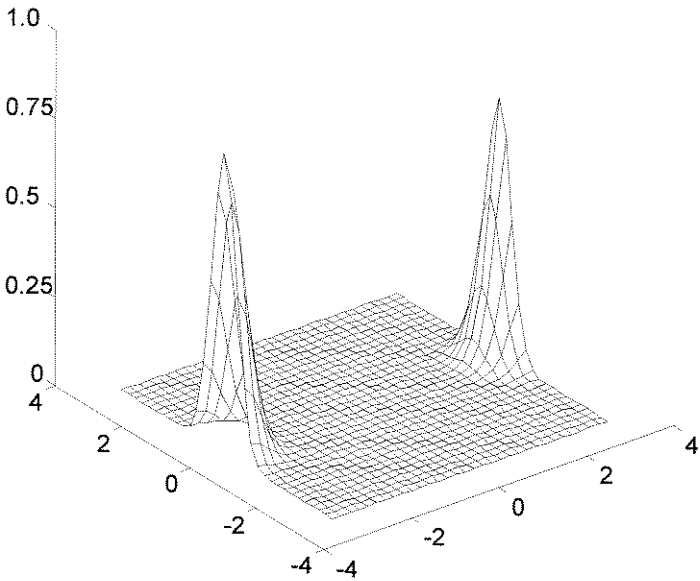


Figura 4.7b – Filtro de Butterworth de 2ª ordem gerado pelo método descrito.

O filtro projetado desta maneira, no entanto, funciona apenas para as linhas e vãos orientados horizontalmente. Felizmente, como foi visto na seção 2.2.2, uma rotação da função no domínio espacial corresponde à rotação de sua representação em frequência por um ângulo correspondente. Portanto, o filtro pode ser utilizado para qualquer orientação desde que seja rotacionado pelo respectivo ângulo.

Como o filtro depende da posição no espaço, o filtro resultante será referido como $H[\omega_m, \omega_n; \theta(m, n)]$ e a sua resposta impulsiva como $h[m, n; \theta(m, n)]$. Esta resposta impulsiva pode ser obtida através da Transformada Inversa de Fourier de $H[\omega_m, \omega_n; \theta(m, n)]$ ou pela rotação de $h(m, n, 0)$ pelo ângulo correspondente e implementada como um filtro FIR (*Finite Impulse Response*) através de uma máscara de tamanho $W \times W$.

4.7 - O algoritmo resultante

As partes mais importantes do algoritmo podem ser colocadas de forma reduzida aqui. Seja $\hat{f}(m, n)$ a estimativa da imagem original a ser obtida. Essa estimativa é obtida através da equação

$$\hat{f}(m, n) = \sum_{k=-W/2}^{W/2} \sum_{l=-W/2}^{W/2} h(k, l; d(m, n)) \cdot f(m-k, n-l). \quad (4.10)$$

Neste ponto, o algoritmo trata de aplicar a cada ponto (m, n) da imagem $f(m, n)$ a ser restaurada, o filtro correspondente à orientação local da linha ou do vão papilar. O filtro é implementado como uma máscara $W \times W$ obtida da forma descrita nas seções anteriores.

Como foi dito, trabalha-se com um número limitado de D direções. Portanto, o conjunto de filtros $h[k, l; d(m, n)]$ se restringe a um conjunto de D filtros, já que a função $d(m, n)$, que “quantiza” as direções, é dada pela equação 4.5, lembrada aqui

$$d(m,n) = \left\lfloor \frac{\theta(m,n)}{\frac{\pi}{D}} + 0.5 \right\rfloor \times \frac{\pi}{D}. \quad (4.5)$$

onde $\lfloor d(m,n) \rfloor$ é o maior inteiro menor que $d(m,n)$. Nota-se que $d(m,n)$ é uma função inteira com domínio $\{d(m,n) \in \mathbb{N} \mid 0 \leq d(m,n) < D\}$. O ângulo $\theta(m,n)$ é dado pela equação 4.4.

$$\theta(m,n) = \tan^{-1} \left(\frac{G_n(m,n)}{G_m(m,n)} \right). \quad (4.4)$$

onde $G_m(m,n)$ e $G_n(m,n)$ são os resultados das aplicações dos operadores gradiente G_m e G_n respectivamente.

Este processamento é feito apenas na região relevante da imagem. A figura 4.8 mostra o resultado de todo o procedimento realizado até esta parte.

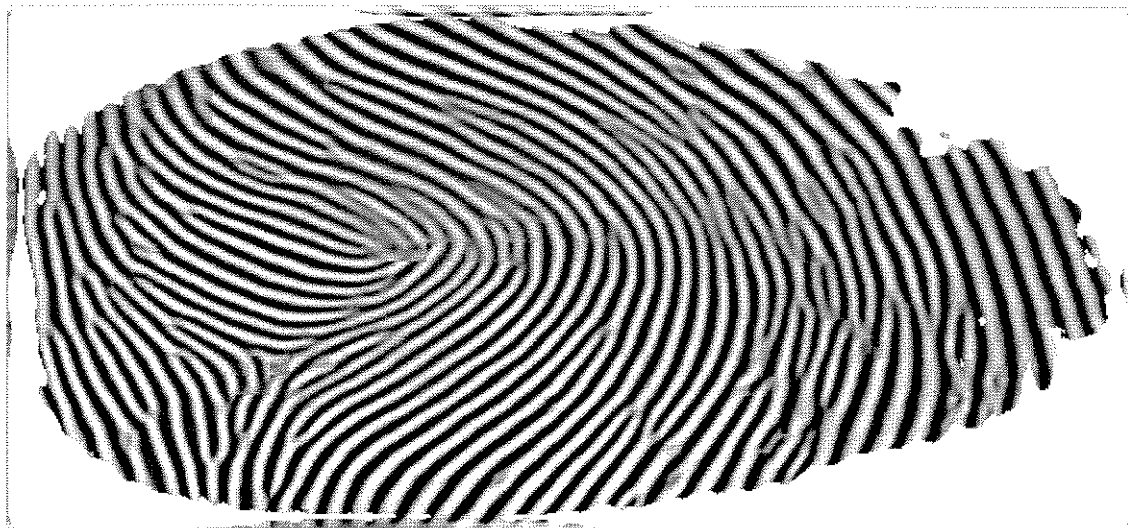


Figura 4.8 - Resultado do algoritmo de filtragem.

4.8 - Limiarização da imagem

Feita a filtragem, o último passo é o mais simples e o mais rápido de todos. Para obter o maior contraste possível, utiliza-se uma das técnicas de *threshold*, ou binarização, descritas na seção 2.3.2.

Embora um *threshold* simples possa ser eficiente para uma impressão digital recuperada por este algoritmo, é preferível utilizar um *threshold* dinâmico. Isso porque a filtragem não garante que informações de frequência muito baixa (*i.e.*: variações no nível do fundo da imagem ocupando um espaço maior que a janela de filtragem) sejam eliminadas. Em outras palavras, pode acontecer que o limiar T bom para uma determinada região da imagem cause distorções em outra.

Portanto, divide-se a imagem em sub-imagens de tamanho $W \times W$ e executa-se a limiarização naquela parte da imagem. O valor do *threshold* T [6] escolhido é a média dos valores dos *pixels* adjacentes naquela região, conforme a equação

$$T = \frac{1}{W^2} \sum_{i=0}^W \sum_{j=0}^W f(m+i, n+j). \quad (4.11)$$

Como a binarização não é o ponto crítico do algoritmo, esse procedimento é suficiente para se obter bons resultados. A figura 4.9 mostra o resultado final do processamento do algoritmo na imagem exemplo, e, para efeito de comparação, é apresentado juntamente com a impressão digital original na figura 4.1.



Figura 4.1 - Imagem exemplo, reproduzida para efeitos de comparação.



Figura 4.9 - Resultado final do processamento.

4.9 - Processamento “em lotes”

Nesta seção descreve-se como executar o algoritmo de maneira seqüencial, ou seja, obtendo os resultados completos de cada passo do algoritmo e depois combinando os resultados para obter a imagem processada.

O procedimento para obter a região relevante da imagem é executado de maneira idêntica, exceto que se o algoritmo for processado desta forma o resultado desse passo é armazenado em um arquivo e funciona como uma “máscara” para eliminar a região não-relevante da imagem.

O procedimento para obter a orientação local também não muda. A orientação local para cada ponto forma uma imagem que deve ser salva em um arquivo. Nesta etapa, sendo o algoritmo executado em lotes, alguns procedimentos podem ser usados para corrigir pontos onde, por ruído ou algum outro motivo, a orientação não tenha sido estimada de maneira correta. Uma técnica eficaz para tanto é o uso do cálculo da relaxação para fazer esta correção.

A outra etapa, de filtragem, é a que mais sofre modificações. Ao invés de adaptar o filtro para cada região filtrada, a imagem como um todo deve ser filtrada na mesma direção. Assim, teremos D imagens $\hat{f}_0(m,n), \hat{f}_1(m,n), \dots, \hat{f}_{D-1}(m,n)$, que serão os resultados das filtrações nas D direções.

Neste ponto, pode-se usar filtrações mais eficazes e complexas para obter imagens com melhores resultados (*e.g.*: fazer a filtragem no domínio da frequência pode fornecer melhores resultados).

Por fim, apenas os pontos relevantes da imagem devem permanecer. Para isso, basta executar uma operação lógica AND com a imagem da região relevante da impressão digital.

$$\hat{f}(m, n) = \hat{f}_{d(m, n)}(m, n) \& r_g(m, n), \quad (4.12)$$

onde $\&$ é o operador AND e $r_g(m, n)$ é a região relevante da imagem.

O grande benefício deste método é a possibilidade de utilizar técnicas mais eficientes para estimar os diversos parâmetros da imagem. As desvantagens são que este método utiliza um espaço de memória muito maior e tende, por isso mesmo, a ser executado mais lentamente.

5. Resultados Obtidos

5.1 - Implementação do algoritmo

Para a implementação do algoritmo e a observação dos resultados, utilizou-se a linguagem C, e o programa foi compilado e rodado em uma estação Sun Sparcstation 4. A implementação teve vários detalhes a serem descritos aqui.

Inicialmente, elaborou-se um programa para calcular as máscaras de gradiente a serem utilizadas. Como foi falado anteriormente, devido à utilização de um modelo idealizado para toda e qualquer impressão digital, não existiu a necessidade de se recalcularem estas máscaras a cada execução do programa. Esse cálculo é bastante complexo e demorado. As máscaras de gradiente foram armazenadas em arquivos e carregadas pelo programa principal quando necessária a sua utilização.

O filtro utilizado para a filtragem direcional foi obtido através da Transformada Inversa de Fourier do filtro de Butterworth de 2^a. ordem conforme descrito no capítulo 4 (ver figura 4.7b). Como trabalhou-se com máscaras de tamanho ímpar, não foi utilizada a Transformada Rápida de Fourier, pois esta exige um número de componentes que seja potência de 2, e a adição de mais dados para completar este número poderia resultar em distorções ao se obter o filtro espacial. Entretanto, assim como as máscaras de gradiente, este filtro é suficiente para qualquer impressão digital, portanto, os filtros espaciais obtidos foram armazenados em arquivos e carregados pelo programa principal quando necessários.

A rotação destes filtros é uma tarefa computacionalmente simples e rápida. Portanto, a obtenção dos D filtros a fazerem a filtragem é feita no momento da execução do algoritmo, sem haver perda significativa de eficiência computacional. Foram utilizadas $D = 16$ direções de filtragem, igualmente espaçadas de $\pi/16$ radianos.

Ao executar o algoritmo, o programa carrega toda a imagem para a memória RAM do computador, e na memória são feitos todos os cálculos e armazenados os resultados do procedimento. Terminados estes cálculos, os resultados são normalizados para o intervalo [0-255] (um byte) e armazenados como uma imagem.

Para avaliar o tempo de duração da execução do algoritmo para cada imagem, utilizou-se funções específicas da linguagem C.

Este capítulo mostra os resultados obtidos com este procedimento, simulando relações sinal-ruído em imagens não deterioradas, e também a *performance* do algoritmo em imagens com as deteriorações típicas de uma impressão digital.

5.2 - Simulações

Para avaliar a *performance* do algoritmo, testar suas várias facetas e observar os resultados, inicialmente se fez os testes em uma imagem considerada boa, deteriorada com ruído aditivo gaussiano com várias relações sinal-ruído. As próximas páginas mostram os resultados obtidos.

A figura 5.1 é a imagem usada para testes. O programa detecta nesta imagem uma largura média das linhas de 13,333333 *pixels*. A imagem tem 337 linhas por 536 colunas. A potência do sinal foi calculada em 14600.

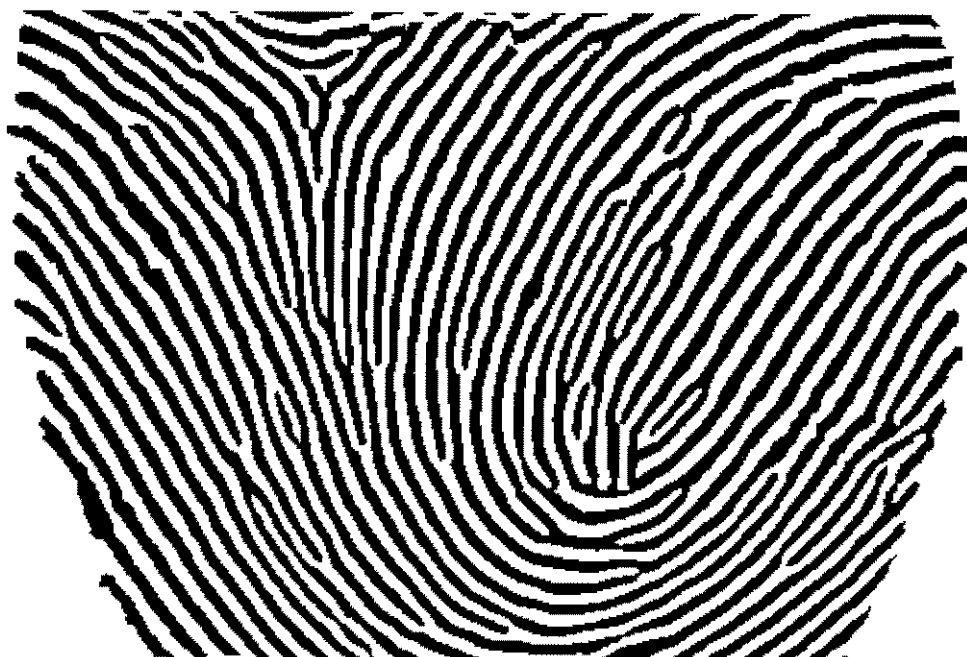


Figura 5.1 - Imagem utilizada para testes.

5.2.1 - Relação Sinal-Ruído igual a 35

A figura 5.2 mostra a imagem-teste deteriorada com ruído aditivo gaussiano com média 127 e desvio-padrão 20 e potência de ruído igual a 400, o que resulta na relação sinal-ruído igual a 35.

Para a figura nestas condições, o algoritmo detectou a largura da linha como sendo 12,825000 *pixels*. A execução do algoritmo durou 2 minutos e 35 segundos. O resultado do processamento está na figura 5.3. A região relevante da imagem está na figura 5.4.

Para o processamento com esta relação sinal-ruído, não foi notada nenhuma falha na imagem com relação a imagem original, no que se refere à identificação da imagem através de suas minúcias.

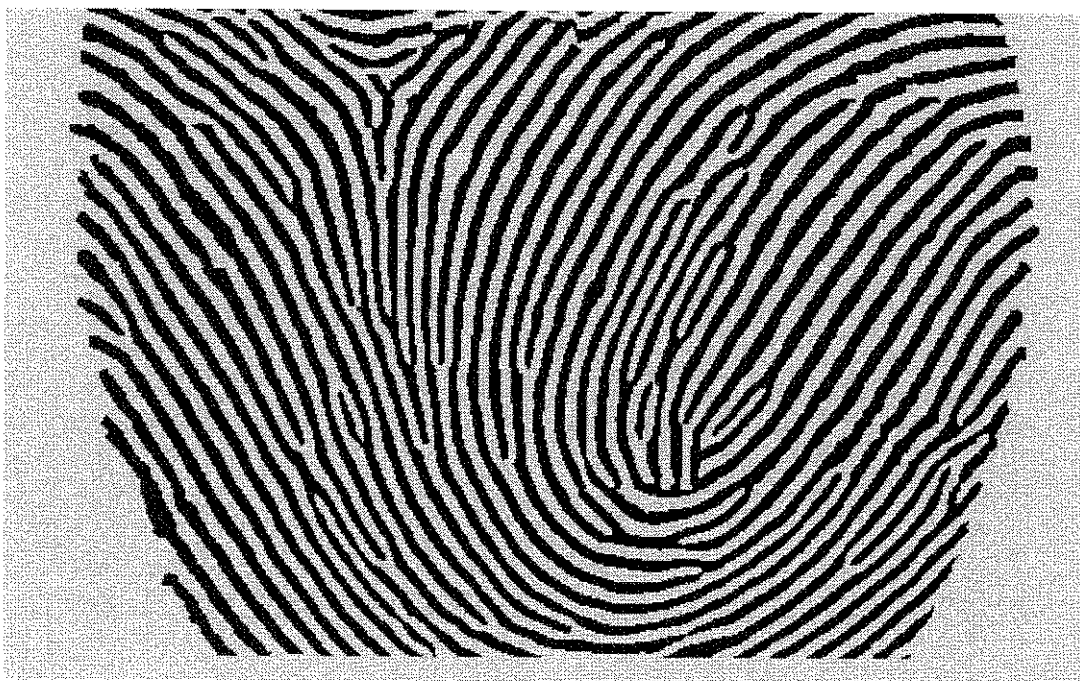


Figura 5.2 - Imagem teste deteriorada com $SNR=35.0$.

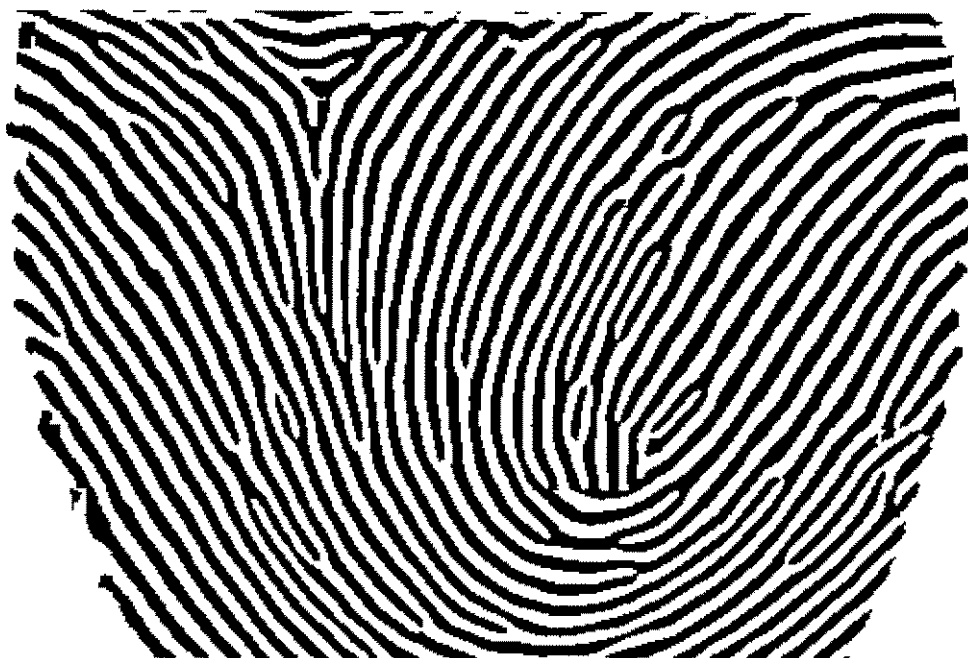


Figura 5.3 - Resultado do processamento da figura 5.2 pelo algoritmo.

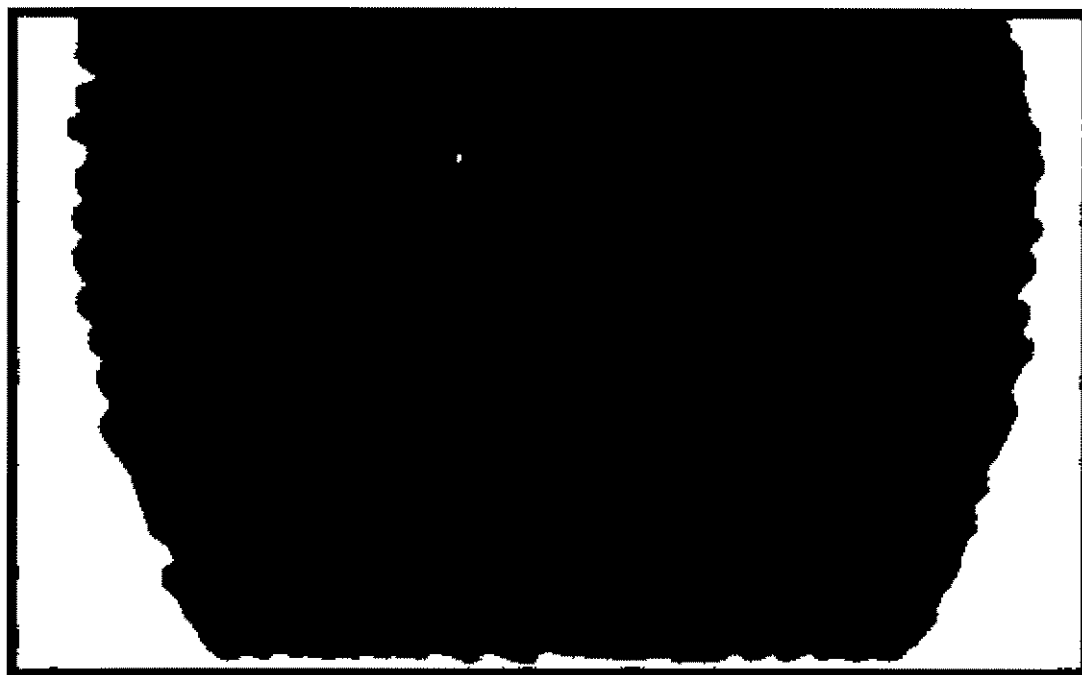


Figura 5.4 - Região relevante da imagem na figura 5.2.

5.2.2 - Relação Sinal-Ruído igual a 0.6

A figura 5.5 mostra a imagem-teste deteriorada com ruído aditivo gaussiano com média 127 e desvio-padrão de 150 e potência de ruído igual a 22500, o que resulta na relação sinal-ruído igual a 0.6.

Para a figura nestas condições, o algoritmo detectou a largura da linha como sendo 13,2312350 *pixels*. A execução do algoritmo durou 2 minutos e 32 segundos. O resultado do processamento está na figura 5.6. A região relevante da imagem está na figura 5.7.

Também para esta imagem não se verificou falhas que possam comprometer o resultado do processamento.

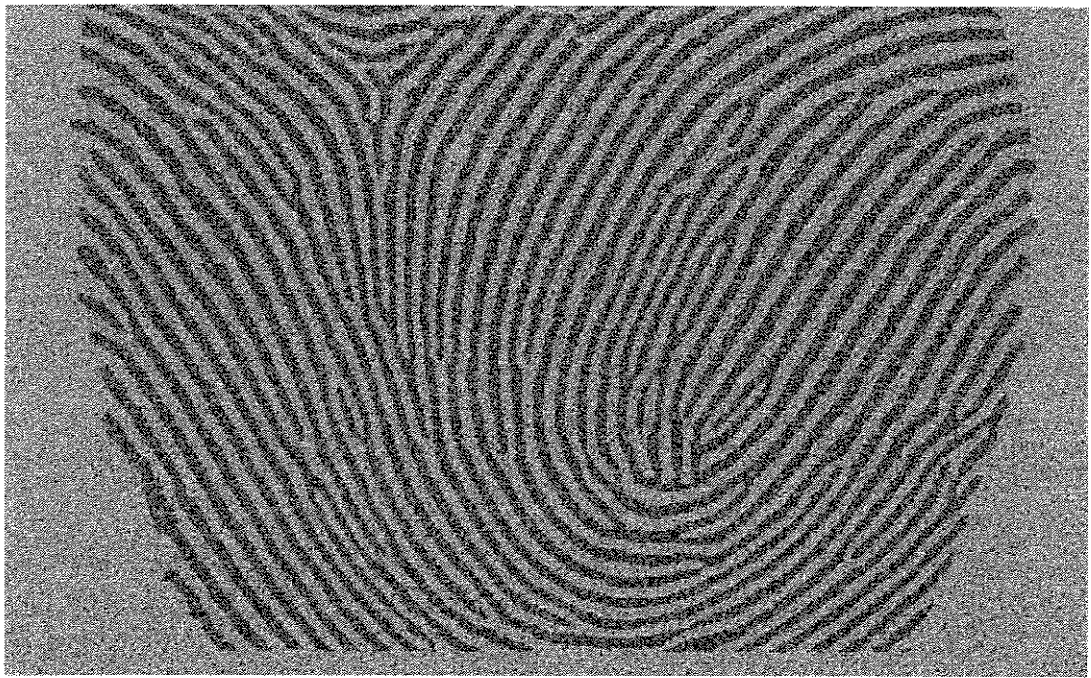


Figura 5.5 - Imagem teste deteriorada com $SNR=0.6$

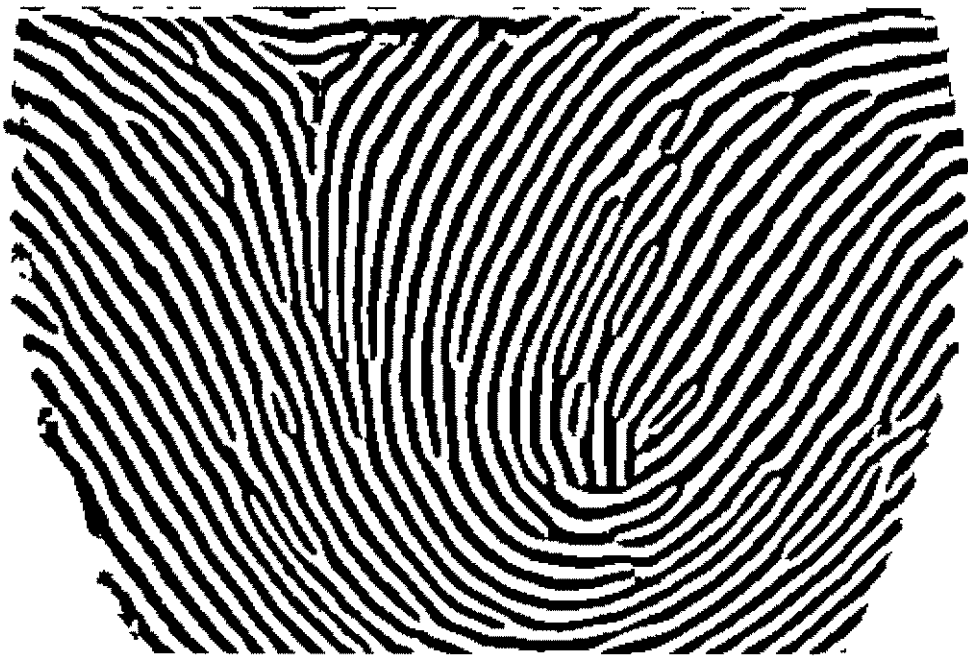


Figura 5.6 - Resultado do processamento da figura 5.5 pelo algoritmo.



Figura 5.7 - Região relevante da imagem na figura 5.5.

5.2.3 - Relação Sinal-Ruído igual a 0.375

A figura 5.8 mostra a imagem-teste deteriorada com ruído aditivo gaussiano com média 127 e desvio-padrão de 200 e potência de ruído igual a 40000, o que resulta em uma relação sinal-ruído igual a 0.375.

Para a figura nestas condições, o algoritmo detectou a largura da linha como sendo 14,118750 *pixels*. A execução do algoritmo durou 2 minutos e 37 segundos. O resultado do processamento está na figura 5.9. A região relevante da imagem está na figura 5.10.

Para esta figura, o resultado do algoritmo começa a apresentar algumas falhas, como é possível notar na figura 5.10. Em alguns pontos, a detecção da região relevante da imagem não foi bem sucedida. Ainda que nesta imagem este fato não tenha gerado distorções muito grandes, em outras imagens isso poderia causar o aparecimento de muitas falsas minúcias, que poderiam significar a falsa identificação da impressão digital.

Entretanto, verificou-se que a região relevante da imagem foi calculada com bastante precisão. Em apenas poucos pontos interiores à impressão digital a estimativa foi feita de maneira errada, bem como nos pontos exteriores à imagem.

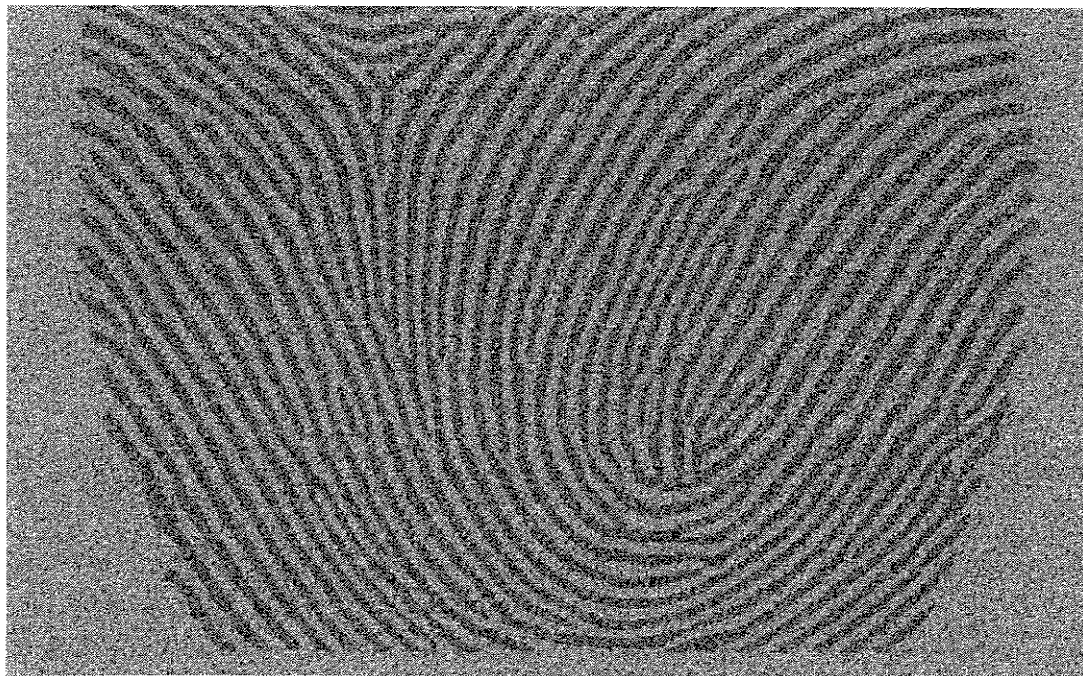


Figura 5.8 - Imagem teste deteriorada com $SNR=0.375$

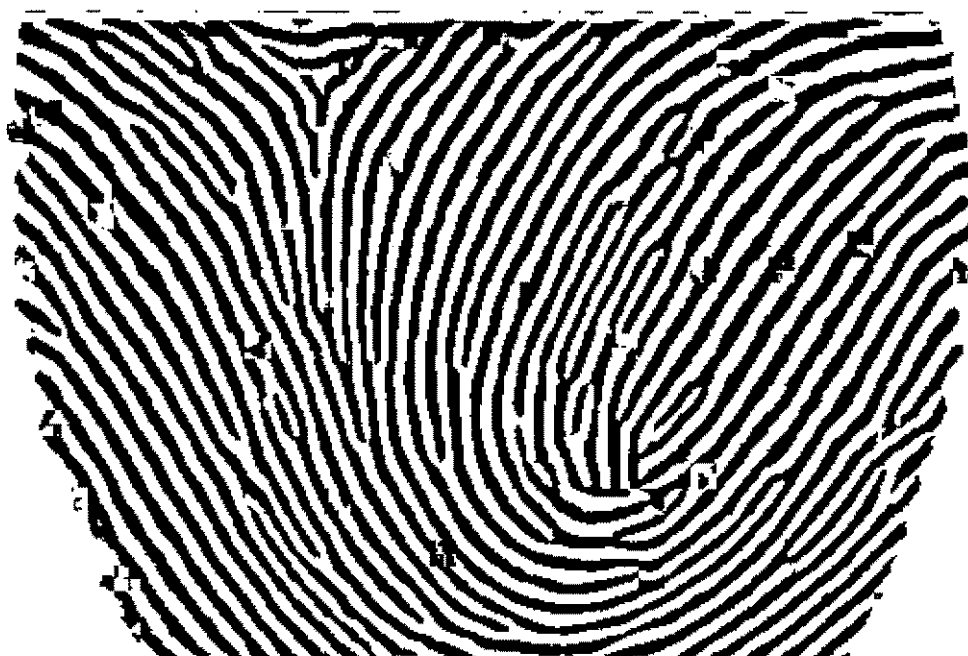


Figura 5.9 - Resultado do processamento da figura 5.8 pelo algoritmo.

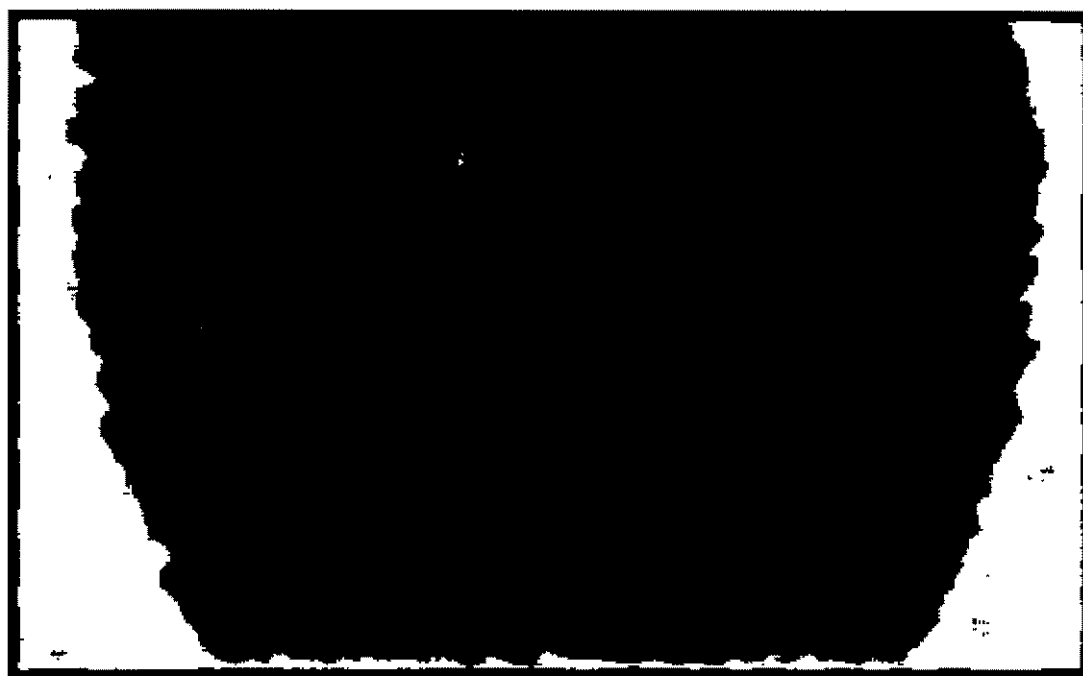


Figura 5.10 - Resultado do processamento do algoritmo na figura 5.8.

5.2.4 - Relação Sinal-Ruído igual a 0.3

A figura 5.11 mostra a imagem-teste deteriorada com ruído aditivo gaussiano com média 127 e desvio-padrão de 220 e potência de ruído igual a 44000, o que resulta na relação sinal-ruído igual a 0.3.

Para a figura nestas condições, o algoritmo detectou a largura da linha como sendo 13,381250 *pixels*. A execução do algoritmo durou 2 minutos e 33 segundos. O resultado do processamento está na figura 5.12. A região relevante da imagem está na figura 5.13.

Na imagem com esta relação sinal-ruído, o resultado foi bem aquém do desejado. A figura apresenta inúmeros pontos onde a estimativa da orientação local não foi precisa, o que ocasionou o aparecimento de várias distorções e várias falsas minúcias. Portanto, para uma imagem com este nível de ruído, o processamento não foi eficiente.

No entanto, a estimativa da região relevante da imagem obteve um resultado bastante satisfatório. Apesar de em vários pontos no interior e no exterior da imagem não terem sido bem sucedidos, a forma da impressão digital ainda pode ser reconhecida com facilidade.

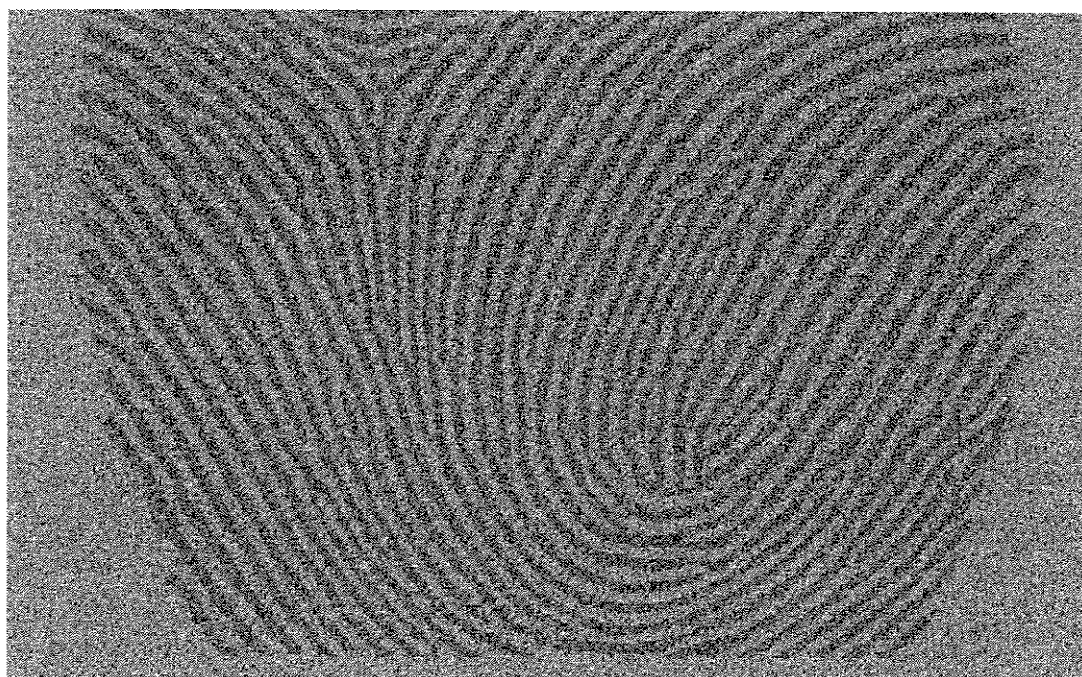


Figura 5.11 - Imagem teste deteriorada com $SNR=0.3$

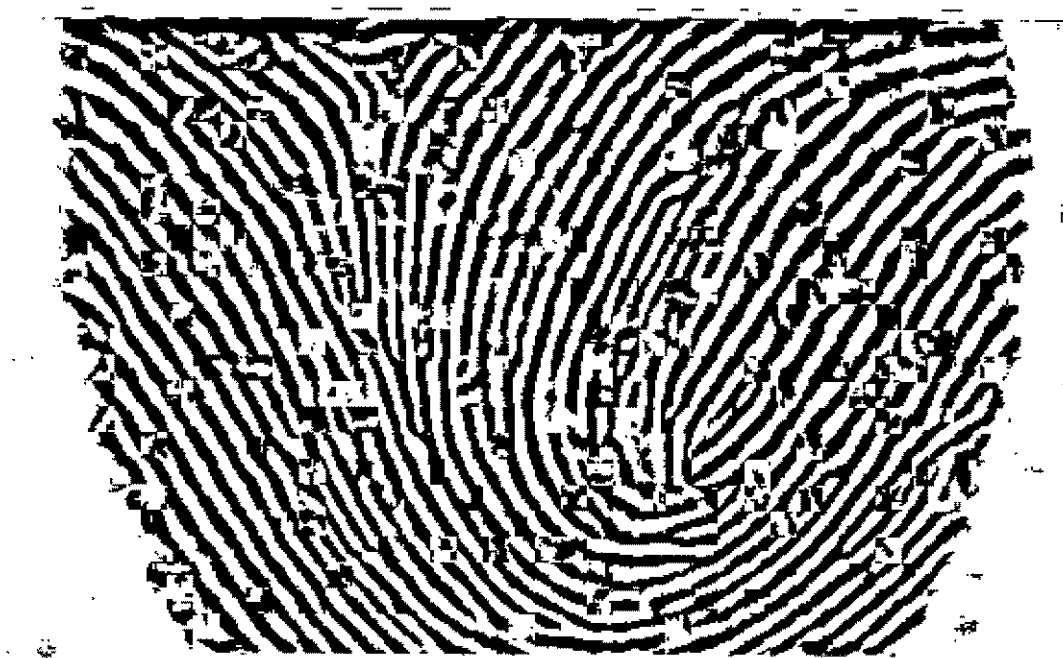


Figura 5.12 - Resultado do processamento da figura 5.11 pelo algoritmo.

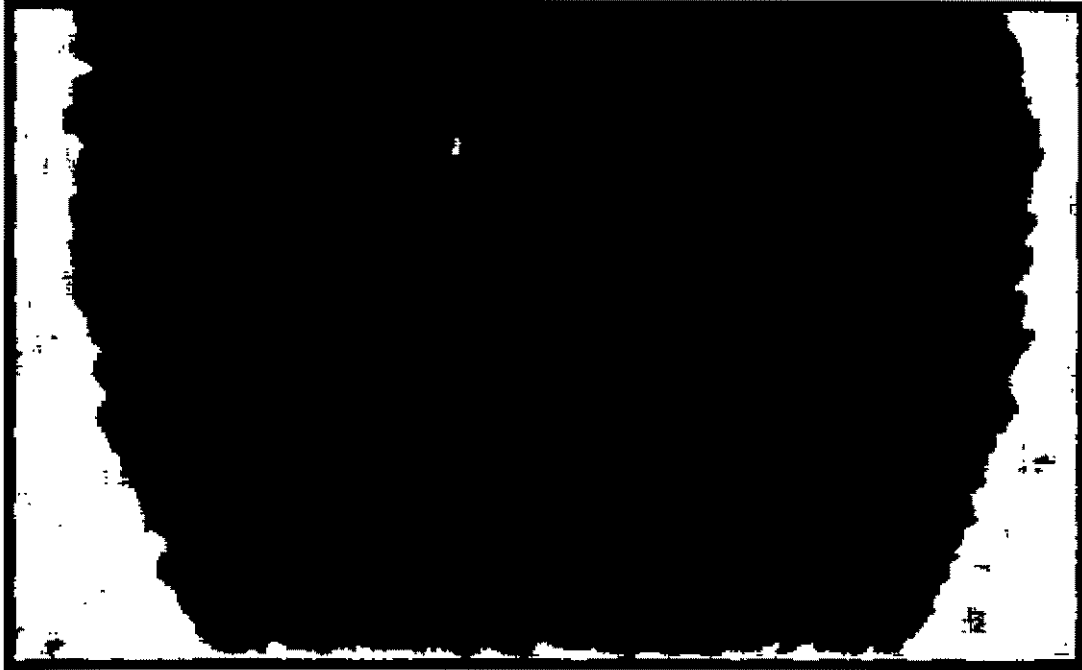


Figura 5.12 - Resultado do processamento do algoritmo na figura 5.11.

5.3 - Restaurações em imagens reais

A restauração em imagens reais possui apenas uma diferença das simulações executadas: o ruído nas imagens reais não é gaussiano, pelo contrário, é altamente correlacionado. Isso se deve ao fato de que o ruído em uma impressão digital é causado basicamente pelo excesso de tinta nos dedos ou pelo deslocamento da mão quando a imagem é adquirida. Dessa forma, o ruído tende a seguir a conformação das linhas papilares.

A seguir, temos algumas imagens reais que foram processadas pelo algoritmo. O método de aquisição de quase todas estas imagens foi obter a impressão digital em um papel em branco com tinta de carimbo e depois digitalizar esta imagem. Em outras imagens, o método foi digitalizar diretamente a impressão digital em um prisma triangular iluminado lateralmente. Cada imagem conterá a informação apropriada a esse respeito.

5.3.1 - Impressão digital em *loop*.

Esta primeira imagem foi obtida utilizando um prisma triangular iluminado lateralmente. A imagem original, que contém 305×587 *pixels*, está mostrada na figura 5.14, o resultado do processamento está ilustrada na figura 5.15 e a figura 5.16 mostra a região relevante desta imagem. A largura das linhas papilares foi estimada em 15,320988 *pixels*. O processamento demorou em média 2 minutos e 24 segundos.

Nesta imagem nota-se o aparecimento de algumas pequenas imperfeições no processamento, principalmente nas bordas da imagem. Isso deve-se ao fato de que o operador gradiente pode interpretar a direção das linhas na borda da imagem de maneira errônea. Outras partes mais borradas na imagem também provocam o aparecimento de falsas minúcias, mas em número pequeno, não comprometendo a eficiência do algoritmo.



Figura 5.14 - Imagem de impressão digital a ser restaurada.



Figura 5.15 - Resultado do processamento da impressão digital da figura 5.14 pelo algoritmo.

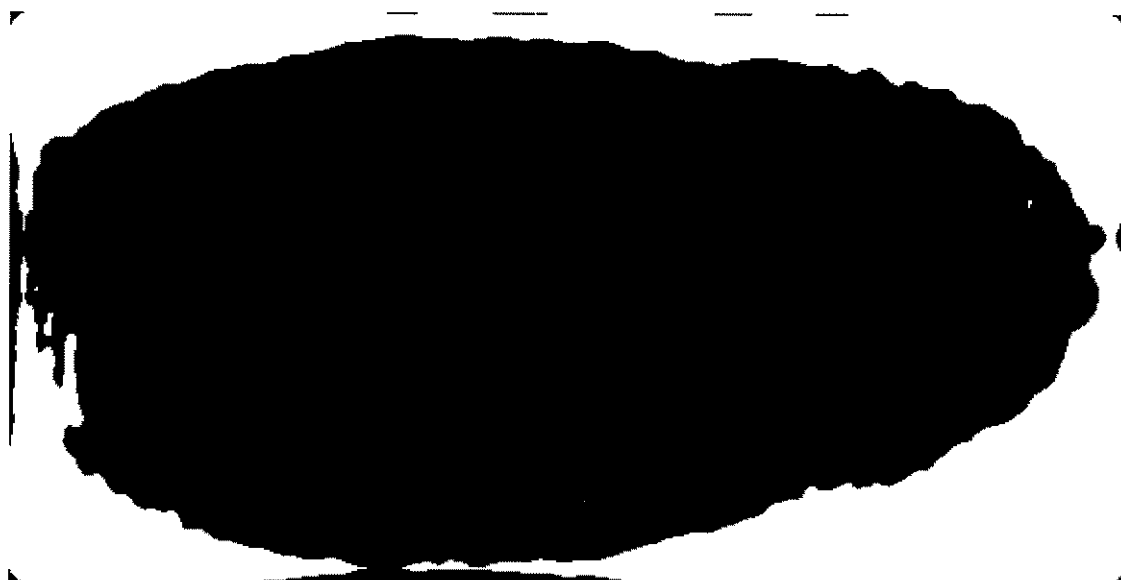


Figura 5.16 - Região relevante da impressão digital da figura 5.14.

5.3.2 - Impressão digital em vórtice.

A imagem a seguir foi obtida pelo mesmo método da imagem mostrada na figura 5.14, ou seja, a partir da iluminação oblíqua de um prisma retangular marcado com a impressão digital. A figura 5.17 contém a imagem original com 581×561 *pixels*, a figura 5.18 contém o resultado do processamento do algoritmo nesta imagem e a figura 5.19 contém a região relevante desta imagem. A largura das linhas papilares foi estimada em $15,647058$ *pixels*. O processamento durou em média 4 minutos e 31 segundos.

Na figura 5.19, nota-se um espaço na região relevante da imagem, causado pela junta do dedo dobrada no momento da aquisição da imagem. Devido aos mesmos fatores da imagem mostrada na seção 5.3.1, as direções nas bordas deste espaço podem ter sido calculadas erroneamente, causando o surgimento de pequenos detalhes que não faziam parte da imagem original. Algumas falsas minúcias apareceram em uma parte significativa da imagem. O motivo para isto é que, nessa região, as linhas tem uma largura bem menor que a largura estimada pelo algoritmo.



Figura 5.17 - Imagem de impressão digital a ser processada.



Figura 5.18 - Resultado do processamento da impressão digital da figura 5.17 pelo algoritmo.

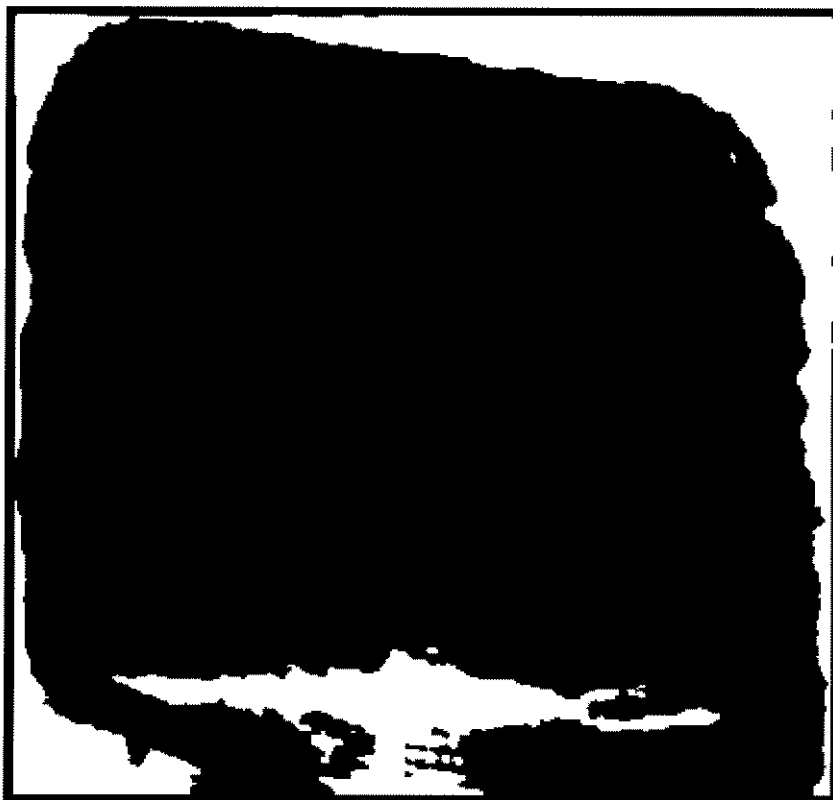


Figura 5.20 - Região relevante da impressão digital na figura 5.17.

5.3.3 - Impressão digital em vórtice.

A imagem a seguir foi obtida marcando-se a impressão digital em papel com tinta de carimbo e digitalizando a imagem. A figura 5.20 contém a imagem original com 477×562 *pixels*, a figura 5.21 contém o resultado do processamento do algoritmo na imagem, e a figura 5.22 contém a região relevante da imagem. A largura das linhas papilares foi estimada em 11.420168 *pixels*. O processamento durou em média 3 minutos e 42 segundos.

Na figura 5.22 nota-se uma ocorrência bem menor de falsas minúcias e outros erros causados pelo processamento. Adicionalmente, a incidência de deteriorações na borda também é bem menor.



Figura 5.20 - Imagem da impressão digital a ser processada.



Figura 5.21 - Resultado do processamento da impressão digital da figura 5.20 pelo algoritmo.

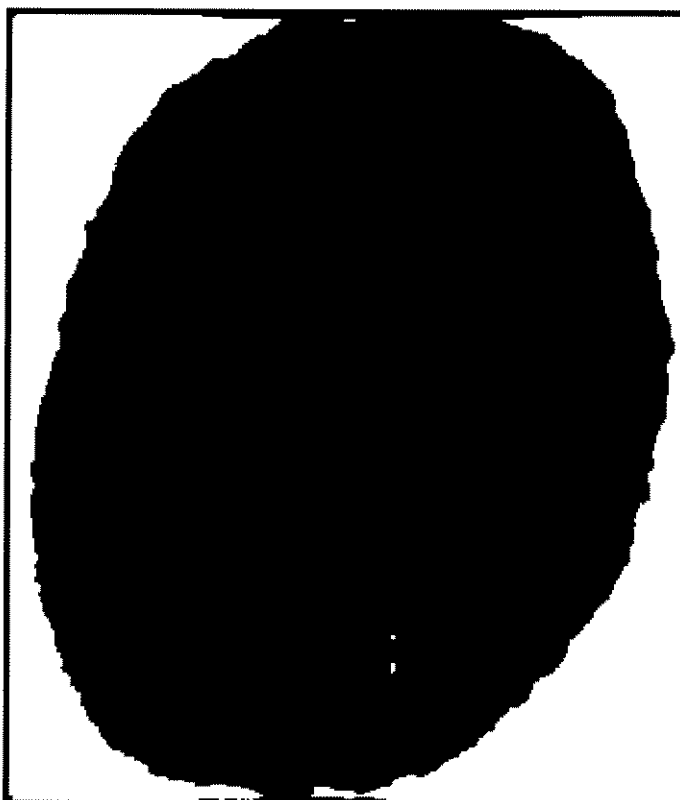


Figura 5.22 - Região relevante da impressão digital na figura 5.20.

5.3.4 - Impressão digital em *loop*.

A imagem mostrada na figura 5.23 foi obtida marcando-se a impressão digital em papel com tinta de carimbo e digitalizando a imagem. A imagem original contém 397×533 *pixels*. A figura 5.24 contém o resultado do processamento da imagem pelo algoritmo e a figura 5.25 contém a região relevante da imagem. A largura das linhas papilares foi estimada em 11.614583 *pixels*. O processamento durou em média 2 minutos e 55 segundos.



Figura 5.23 - Imagem da impressão digital a ser processada.

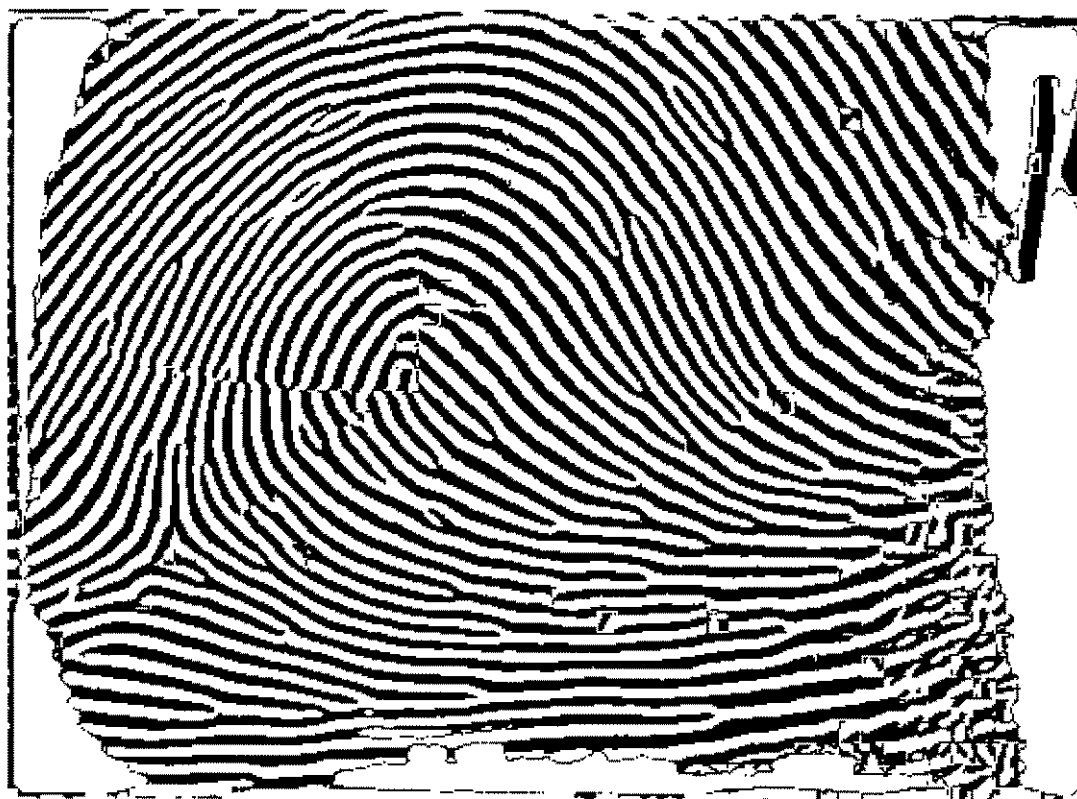


Figura 5.24 - Resultado do processamento da impressão digital da figura 5.23 pelo algoritmo.



Figura 5.25 - Região relevante da impressão digital da figura 5.23.

5.4 - Um contra-exemplo

Embora o algoritmo tenha funcionado bem e obtido excelentes resultados para a maior parte das imagens em que foi testado, nem todas as imagens forneceram um bom resultado em seu processamento.

A imagem mostrada na figura 5.26 tem 556×521 *pixels*, e foi obtida com tinta de carimbo. Nesta imagem original, podemos notar na porção superior esquerda que houve excesso de tinta ou talvez o deslocamento do dedo durante a aquisição da imagem. O resultado é que a imagem ficou borrada naquela região, apagando praticamente todo vestígio de linhas e vãos.

A largura média da linha papilar foi estimada em 15.202206 *pixels*, e o processamento durou em média 4 minutos e 17 segundos. O resultado do processamento é mostrado na figura 5.27, e a região relevante da imagem está ilustrada na figura 5.28.

Como pode-se notar facilmente, apareceram várias falsas minúcias nesta imagem, principalmente na região mencionada. Isso se deve ao fato de que, com o excesso de ruído naquela região (tanto que obscurecem os dados relevantes da imagem), tornou-se impossível para o operador gradiente detectar corretamente a direção das linhas papilares. A região relevante foi calculada com alguma precisão, também através do operador gradiente. Essa maior eficiência se deve ao fato de que detectar as linhas exige uma menor precisão de cálculo que detectar a direção que elas seguem na mesma região. Ainda assim, em vários pontos desta região a direção das linhas foi estimada corretamente.



Figura 5.26 - Imagem de impressão digital a ser processada.



Figura 5.28 - Resultado do processamento da impressão digital da figura 5.26 pelo algoritmo.



Figura 5.28 - Região relevante da imagem da figura 5.26.

6. Conclusão

Neste capítulo, analisamos a validade do trabalho, a eficiência do algoritmo obtido e suas restrições, bem como as contribuições e sugestões para pesquisas futuras.

6.1 - Análise dos resultados obtidos

Como pode-se verificar a partir dos resultados demonstrados no capítulo 5, o algoritmo funciona de maneira satisfatória para boa parte dos casos reais de impressões digitais corrompidas por ruído. Os testes realizados mostraram que até para uma relação sinal-ruído significativamente baixa é possível restaurar uma imagem com bastante precisão.

Também foi possível concluir que o algoritmo é executado de maneira bastante rápida. Mesmo com imagens grandes (cerca de 200.000 a 300.000 pontos a serem processados) o tempo de processamento foi bastante pequeno, tendo sido a maior imagem processada em menos de cinco minutos, com processador não-dedicado e implementação não otimizada para maior velocidade. Considerando os resultados obtidos, concluímos que o algoritmo é executado de maneira rápida e eficiente.

Alguns erros e falsas minúcias foram geradas, mas a identificação não fica prejudicada. Esses pequenos erros devem-se basicamente a dois fatores principais (a seção 6.3 oferece sugestões para sua solução):

- O período local (*i.e.*: a largura das linhas e dos vãos) é significativamente maior ou menor que a janela usada. Comprovou-se que a largura local pode variar significativamente pela imagem da impressão digital. Se a largura for maior que a janela, a filtragem passa-alta feita na direção perpendicular à propagação das linhas vai eliminar as baixas frequências nesta direção, e portanto pode criar detalhes que não fazem parte da imagem. Se a largura for menor, o operador gradiente pode confundir-se com relação à direção de propagação das linhas, dessa forma criando falsas minúcias.

- O ruído na imagem da impressão digital é altamente correlacionado, pois decorre em grande parte do deslizamento do dedo na superfície. Dessa forma, o ruído varia de uma região para outra, podendo haver regiões em que o ruído é tão grande que se torna impossível para o operador gradiente detectar, naquele ponto, a direção de propagação das linhas. Nessas regiões, a restauração pode ser feita considerando uma direção errada, prejudicando a restauração.

6.2 - Contribuições do trabalho

O trabalho fornece duas contribuições importantes. A primeira é o estudo matemático da impressão digital conforme feito no capítulo 3. A segunda e mais significativa contribuição é o algoritmo derivado desse estudo.

O estudo e o modelamento matemático descrito no capítulo 3 forneceram um modo de equacionar as características da impressão digital, possibilitando, dessa forma, que outros estudos visando outros objetivos ou um refinamento deste algoritmo sejam desenvolvidos. Ainda que o modelo desenvolvido não seja completo, ele é flexível, e é possível estendê-lo para abranger outras características como a largura local, o nível de ruído local, a curvatura local das linhas, entre outras. Equacionando estas e outras características é possível elaborar um algoritmo que leve em conta outros detalhes da impressão digital e beneficie outros estudos relacionados, como por exemplo a identificação.

A segunda contribuição deste trabalho, o algoritmo em si, é um método rápido e eficiente de enriquecer a impressão digital e facilitar a sua identificação. Para os peritos, diminui a confusão visual causada pelos *borrões* da impressão digital. Para o reconhecimento computacional, o pré-processamento da imagem pelo algoritmo torna a identificação, geralmente feita por métodos demorados, mais simples, pois as minúcias usadas para a identificação são mais facilmente ressaltadas.

6.3 - Sugestões para trabalhos futuros

Várias sugestões podem ser feitas para dar continuidade a este trabalho. Entre os mais importantes estão os refinamentos a o algoritmo. Este capítulo sugere alguns desses aprimoramentos.

A principal sugestão é incluir novas características no equacionamento da impressão digital e refazer o algoritmo para abranger essas alterações. Os problemas citados na seção 6.1 podem ser resolvidos modificando o modelo do capítulo 3. Entre estas alterações pode-se incluir a largura local das linhas, o nível de ruído local, a direção do deslizamento do dedo no momento da aquisição, a curvatura local das linhas, entre outros. Modificar o modelo para incluir bifurcações e interrupções também é um caminho a ser seguido.

Outra sugestão é levar em conta as características de pontos adjacentes ao se calcular as estatísticas de cada ponto. Por exemplo, utilizar algum método que considere a direção local de pontos adjacentes aumenta a eficiência da detecção. O uso do método da relaxação equacionado corretamente permitirá separar as regiões da imagem onde os pontos têm direções semelhantes.

Adaptar outros elementos além da direção da filtragem a cada ponto é outra opção. Por exemplo, estimar a cada ponto ou grupo de pontos o nível de ruído local permitirá adaptar o operador gradiente estocástico para trabalhar de maneira mais eficiente a cada ponto. Outro exemplo é estimar a largura local a cada ponto e modificar o tamanho da janela para estar em conformidade com esta estimativa.

Finalmente, devem ser testados outros tipos de implementações, como, por exemplo, utilizar um microprocessador especializado em Processamento de Sinais (como o Motorola 56000) para implementar o algoritmo. Tal implementação possivelmente resultará em um processamento mais rápido, já que o processador é dedicado e especializado, e também possibilitará o uso do algoritmo em sistemas automáticos de identificação.

Apêndice 1 - Máscaras de Gradiente utilizadas

Este apêndice contém as máscaras de gradiente utilizadas para a implementação do algoritmo. Estas máscaras foram calculadas resolvendo a equação 2.20, com relação sinal-ruído de 0.5.

Para calcular os coeficientes, utilizou-se a técnica de matriz em blocos. A ordenação dos elementos em vetores foi feita por linha. Maiores referências sobre o assunto podem ser encontradas em [4].

O cálculo foi feito utilizando o MatLab versão 4.0, e o resultado é apresentado no formato típico utilizado por este aplicativo por conveniência. As características de simetria do operador gradiente também foram exploradas para reduzir o número de coeficientes a serem calculados. São mostrados apenas os resultados dos coeficientes calculados, sendo que os outros podem ser obtidos através da relação:

$$G_{ij} = G_{(n-i)(n-j)} \quad (\text{eq. A1.1})$$

onde n é a dimensão da máscara. São mostradas apenas as máscaras G_m . As máscaras G_n podem ser obtidas da transposta de G_m .

A1.1 - Máscara 3×3

-0.3333	0
-1.0000	0

A1.2 - Máscara 5×5

-0.2000	-0.2000	0
-0.8000	-0.8000	0
-1.0000	-1.0000	0

A1.3 - Máscara 7×7

-0.0385	-0.0769	-0.0769	0
-0.1923	-0.3846	-0.3846	0
-0.3846	-0.7692	-0.7692	0
-0.5000	-1.0000	-1.0000	0

A1.4 - Máscara 9×9

-0.0061	-0.0182	-0.0303	-0.0242	0
-0.0364	-0.1091	-0.1818	-0.1455	0
-0.0970	-0.2909	-0.4848	-0.3879	0
-0.1697	-0.5091	-0.8485	-0.6788	0
-0.2000	-0.6000	-1.0000	-0.8000	0

A1.5 - Máscara 11×11

-0.0009	-0.0037	-0.0084	-0.0112	-0.0084	0
-0.0066	-0.0262	-0.0590	-0.0787	-0.0590	0
-0.0215	-0.0861	-0.1938	-0.2584	-0.1938	0
-0.0468	-0.1873	-0.4213	-0.5618	-0.4213	0
-0.0721	-0.2884	-0.6489	-0.8652	-0.6489	0
-0.0833	-0.3333	-0.7500	-1.0000	-0.7500	0

A1.6 - Máscara 13×13

-0.0001	-0.0007	-0.0019	-0.0034	-0.0041	-0.0029	0
-0.0011	-0.0055	-0.0154	-0.0274	-0.0329	-0.0230	0
-0.0043	-0.0213	-0.0595	-0.1063	-0.1276	-0.0893	0
-0.0110	-0.0549	-0.1536	-0.2743	-0.3292	-0.2305	0
-0.0206	-0.1029	-0.2881	-0.5144	-0.6173	-0.4321	0
-0.0296	-0.1481	-0.4148	-0.7407	-0.8889	-0.6222	0
-0.0333	-0.1667	-0.4667	-0.8333	-1.0000	-0.7000	0

A1.7 - Máscara 15×15

-0.0000	-0.0001	-0.0004	-0.0009	-0.0013	-0.0015	-0.0010	0
-0.0002	-0.0011	-0.0035	-0.0077	-0.0121	-0.0133	-0.0089	0
-0.0008	-0.0047	-0.0156	-0.0343	-0.0538	-0.0593	-0.0398	0
-0.0023	-0.0139	-0.0464	-0.1021	-0.1601	-0.1763	-0.1183	0
-0.0051	-0.0305	-0.1018	-0.2239	-0.3511	-0.3867	-0.2595	0
-0.0087	-0.0522	-0.1739	-0.3825	-0.5999	-0.6607	-0.4434	0
-0.0119	-0.0712	-0.2374	-0.5223	-0.8191	-0.9022	-0.6054	0
-0.0132	-0.0789	-0.2632	-0.5789	-0.9079	-1.0000	-0.6711	0

A1.8 - Máscara 17×17

-0.0000	-0.0000	-0.0001	-0.0002	-0.0004	-0.0005	-0.0005	-0.0003	0
-0.0000	-0.0002	-0.0007	-0.0019	-0.0036	-0.0051	-0.0053	-0.0034	0
-0.0001	-0.0009	-0.0036	-0.0094	-0.0179	-0.0255	-0.0264	-0.0171	0
-0.0005	-0.0032	-0.0122	-0.0317	-0.0602	-0.0856	-0.0887	-0.0575	0
-0.0011	-0.0079	-0.0306	-0.0792	-0.1506	-0.2139	-0.2219	-0.1438	0
-0.0022	-0.0156	-0.0601	-0.1558	-0.2961	-0.4208	-0.4363	-0.2827	0
-0.0035	-0.0248	-0.0958	-0.2483	-0.4717	-0.6704	-0.6952	-0.4505	0
-0.0047	-0.0326	-0.1259	-0.3264	-0.6201	-0.8813	-0.9139	-0.5922	0
-0.0051	-0.0357	-0.1378	-0.3571	-0.6786	-0.9643	-1.0000	-0.6480	0

A1.9 - Máscara 19×19

-0.0000	-0.0000	-0.0000	-0.0000	-0.0001	-0.0001	-0.0002	-0.0002	-0.0001	0
-0.0000	-0.0000	-0.0001	-0.0004	-0.0009	-0.0016	-0.0021	-0.0020	-0.0013	0
-0.0000	-0.0002	-0.0008	-0.0023	-0.0051	-0.0086	-0.0114	-0.0113	-0.0071	0
-0.0001	-0.0007	-0.0029	-0.0086	-0.0189	-0.0322	-0.0426	-0.0421	-0.0266	0
-0.0002	-0.0018	-0.0081	-0.0239	-0.0529	-0.0902	-0.1192	-0.1178	-0.0743	0
-0.0005	-0.0041	-0.0178	-0.0530	-0.1173	-0.1999	-0.2642	-0.2611	-0.1647	0
-0.0009	-0.0074	-0.0323	-0.0961	-0.2125	-0.3622	-0.4786	-0.4731	-0.2984	0
-0.0014	-0.0112	-0.0489	-0.1452	-0.3212	-0.5474	-0.7233	-0.7150	-0.4510	0
-0.0018	-0.0142	-0.0623	-0.1852	-0.4097	-0.6982	-0.9227	-0.9120	-0.5753	0
-0.0019	-0.0154	-0.0676	-0.2008	-0.4440	-0.7568	-1.0000	-0.9884	-0.6236	0

A1.10 - Máscara 21×21

-0.0000	-0.0000	-0.0000	-0.0000	-0.0000	-0.0000	-0.0001	-0.0001	-0.0001	-0.0000	0
-0.0000	-0.0000	-0.0000	-0.0001	-0.0002	-0.0004	-0.0006	-0.0008	-0.0007	-0.0005	0
-0.0000	-0.0000	-0.0001	-0.0005	-0.0012	-0.0024	-0.0038	-0.0048	-0.0046	-0.0028	0
-0.0000	-0.0001	-0.0006	-0.0020	-0.0051	-0.0101	-0.0158	-0.0197	-0.0187	-0.0116	0
-0.0000	-0.0004	-0.0019	-0.0063	-0.0158	-0.0311	-0.0488	-0.0609	-0.0579	-0.0357	0
-0.0001	-0.0009	-0.0046	-0.0155	-0.0388	-0.0764	-0.1200	-0.1497	-0.1424	-0.0879	0
-0.0002	-0.0019	-0.0094	-0.0313	-0.0786	-0.1547	-0.2429	-0.3029	-0.2882	-0.1779	0
-0.0004	-0.0033	-0.0159	-0.0533	-0.1337	-0.2631	-0.4131	-0.5153	-0.4903	-0.3026	0
-0.0005	-0.0047	-0.0231	-0.0772	-0.1938	-0.3812	-0.5986	-0.7467	-0.7105	-0.4385	0
-0.0007	-0.0059	-0.0288	-0.0961	-0.2413	-0.4748	-0.7455	-0.9299	-0.8848	-0.5461	0
-0.0007	-0.0063	-0.0309	-0.1034	-0.2595	-0.5105	-0.8017	-1.0000	-0.9515	-0.5872	0

Bibliografia

- [1] O’Gorman, L. & Nickerson, J.V.: “An Approach to Fingerprint Filter Design”, *Pattern Recognition*, vol. 22, no. 1, 1994, pgs. 29-38.
- [2] Hung, D.C.D: “Enhancement and Feature Purification of Fingerprint Images”, *Pattern Recognition*, vol. 26, no. 11, 1989, pgs 1661-1671.
- [3] Sherlock, B.G.; Monro, D.M. & Millard, K.: “Fingerprint Enhancement by Directional Fourier Filtering”, *IEEE Proc.-Vis. Image Signal Processing*, vol. 141, no. 2, pgs 87-94.
- [4] Gonzales, R.C. & Woods, R.E.: “Digital Image Processing”, World Students Series, Addison-Wesley Publishing Company, 1992.
- [5] Jain, A.K.: “Fundamentals of Digital Image Processing”, Prentice-Hall Information & System Sciences Series, Prentice-Hall, 1989.
- [6] Lim, J.S.: “Two-Dimensional Signal and Image Processing”, Prentice-Hall Processing Series, Prentice-Hall, 1990.
- [7] Parker, J.R.: “Algorithms for Image Processing and Computer Vision”, Wiley Computer Publishing, 1997.
- [8] Daetwiler, J.: “Impressões Digitais Latentes”, *Polícia Técnica*, Editora Abril, 1974, pgs. 49-52.
- [9] Daetwiler, J.: “Impressões Palmares”, *Polícia Técnica*, Editora Abril, 1974, pgs. 117-120.
- [10] Daetwiler, J.: “Sinais Individuais”, *Polícia Técnica*, Editora Abril, 1974, pgs. 175-176.

- [11] Kamijo, M.; Mieno, H. & Kojima, K.: "Classification of Fingerprint Images using a Neural Network", *Systems and Computers in Japan*, vol. 23, no. 3, 1992, pgs 89-101.
- [12] Kass, M & Witkin, A.: "Analyzing Oriented Patterns", *Computer Vision, Graphics and Image Processing*, vol. 37, 1987, pgs. 362-385.
- [13] Knutsson, H.E.; Wilson, R. & Granlund. G.H.: "Anisotropic Nonstationary Image Estimation and its Applications: Part I - Restoration of Noisy Images", *IEEE Transactions on Communications*, COM-31, 1983, pgs. 388-397.
- [14] Niblack, W.: "An Introduction to Digital Image Processing", Prentice-Hall International, 1989.
- [15] Oppenheim, A.V. & Schaffer, R.W.: "Discrete-time Signal Processing", Prentice-Hall Processing Series, Prentice-Hall, 1989.
- [16] Rao, M.: "Feature Extraction for Fingerprint Classification", *Pattern Recognition*, vol. 8, 1976, pgs 181-192.
- [17] Verma, M.R.; Majumdar, A.K. & Chatterjee, B.: "Edge Detection in Fingerprints", *Pattern Recognition*, vol. 20, no. 5, 1987, pgs 513-523.
- [18] Xiao, Q. & Raafat, H.: "Fingerprint Image Postprocessing: a Combined Statistical and Structural Approach", *Pattern Recognition*, vol. 24, no. 10, 1991, pgs. 985-992.