

**UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**Metodologia para extração de características invariantes à
rotação em Imagens de Impressões Digitais**

Cristina Mônica Dornelas Mazetti

Dissertação apresentada à Escola de Engenharia de São Carlos, da Universidade de São Paulo, para obtenção do título de Mestre em Engenharia Elétrica.

São Carlos
2006

Cristina Mônica Dornelas Mazetti

**Metodologia para extração de características invariantes à
rotação em Imagens de Impressões Digitais**

Dissertação apresentada à Escola de
Engenharia de São Carlos, da
Universidade de São Paulo, para
obtenção do título de Mestre em
Engenharia Elétrica.

Orientador: Prof.Dr.Adilson Gonzaga

São Carlos
2006

AUTORIZO A REPRODUÇÃO E DIVULGAÇÃO TOTAL OU PARCIAL DESTE TRABALHO, POR QUALQUER MEIO CONVENCIONAL OU ELETRÔNICO, PARA FINS DE ESTUDO E PESQUISA, DESDE QUE CITADA A FONTE.

Ficha catalográfica preparada pela Seção de Tratamento
da Informação do Serviço de Biblioteca – EESC/USP

M476m Mazetti, Cristina Mônica Dornelas
Metodologia para extração de características
invariantes à rotação em impressões digitais / Cristina
Mônica Dornelas Mazetti ; orientador Adilson Gonzaga. --
São Carlos, 2006.

Dissertação (Mestrado-Programa de Pós-Graduação em
Engenharia Elétrica. Área de Concentração: Processamento
de Sinais e Instrumentação) -- Escola de Engenharia de
São Carlos da Universidade de São Paulo, 2006.

1. Impressão digital. 2. Invariância com relação à
rotação. 3. Reconhecimento de padrões. I. Título.

Ao meu marido, Ademir, e
meus pais, Venina e Cláudio

Agradecimentos

A Deus, pela minha existência, porque nada nos é possível se não for de Sua vontade.

Ao amigo e orientador Prof. Dr. Adilson Gonzaga, pelo constante incentivo, apoio, pelos conselhos, pela mão amiga e principalmente pela confiança que depositou em mim durante este período. Tenho convicção de que fui uma aluna de sorte por ter encontrado em meu caminho uma pessoa que acima de tudo mostrou que nada é impossível quando existe vontade e dedicação. Você é insubstituível.

Ao meu marido, sempre presente, sempre incentivando, sempre com paciência e principalmente com muito amor.

Aos meus pais pela compreensão e apoio, por estarem presentes em mais uma etapa da minha vida. Vocês são os cúmplices de todas as dificuldades que encontramos neste período e são também vencedores e merecedores de todo meu respeito.

Ao meu chefe, Gilson Fernando Laforga pelo apoio, incentivo e dispensa dos horários de trabalho.

A minhas amigas professoras, Edna, Rita e Valéria que de alguma forma colaboraram no desenvolvimento deste trabalho, trocando idéias e opiniões e na revisão do mesmo; todos de alguma forma me ajudaram a gerar este material.

Tenho certeza que agradei apenas uma pequena fração das pessoas a quem devo muito e peço especial perdão àquelas que omiti.

Sumário

Sumário	i
Índice de Figuras	iii
Índice de Tabelas	vii
Lista de Abreviaturas	viii
Resumo	ix
<i>Abstract</i>	x
 Capítulo 1 - Introdução	 01
1.1. Considerações Iniciais	01
1.2. Biometria	02
1.3. Estrutura da Dissertação	07
 Capítulo 2 – Impressão Digital	 08
2.1. Introdução	08
2.2. Conceitos Biométricos para Impressão Digital	09
2.3. Análise e Representação das Impressões Digitais	10
2.3.1. Pontos Singulares (Núcleos e Deltas)	11
2.3.2. Minúcias ou “Detalhes de Galton”	12
2.4. Classificação Exclusiva X Classificação Contínua	13
2.5. Considerações Finais	15
 Capítulo 3 – Extração de Características	 17
3.1. Introdução	17
3.2. Pré-processamento	17
3.2.1. Detecção de Bordas pelo Detector de “ <i>Canny</i> ”	18
3.3. Extração de Aspectos	19
3.4. Número de Cruzamentos (<i>Crossing Number</i> – <i>CN</i>)	19
3.5. Direção da Crista Local	21
3.5.1. Direção da Crista Local do Tipo Final	21
3.5.2. Direção da Crista Local do Tipo Bifurcada	22

3.6. Comparação das Minúcias.....	23
3.6.1. Distância Euclidiana.....	24
3.6.2. Centro de Massa.....	24
3.7. Considerações Finais.....	25
 Capítulo 4 – Métodos para o reconhecimento que consideram a rotação de Impressão Digital	 26
4.1. Introdução	26
4.2. Algumas Técnicas Empregadas	26
4.3. Considerações Finais.....	32
 Capítulo 5 – Metodologia	 33
5.1. Metodologia.....	33
5.1.1. Pré-processamento da Imagem	33
5.1.2. Extração de Minúcias	34
5.1.3. Proposta do Algoritmo de extração de características Invariantes com relação à Rotação	 35
5.1.4. Geração do Vetor de Característica das minúcias	38
5.1.5. Determinação do Vetor Alinhamento	40
5.1.6. Minúcias Espúrias	46
 Capítulo 6 – Discussão dos Resultados.....	 50
6.1. Discussão dos Resultados	50
6.2. Conclusões	64
6.3. Perspectivas Futuras.....	66
 Referências Bibliográficas	 67
 Apêndice A	 71
Apêndice B	77

Índice de Figuras

Figura 1.1. Alguns identificadores biométricos que são usados em várias aplicações.....	02
Figura 1.2. Diagramas das tarefas de registro, autenticação e de identificação.....	04
Figura 2.1. Métodos de Aquisição da Imagem de Impressão Digital	09
Figura 2.2. Cristas e Vales	10
Figura 2.3. Área padrão e tipos de linhas	11
Figura 2.4. Singularidades, geralmente classificadas em Núcleo e Delta	11
Figura 2.5. Aspectos básicos e Aspectos Compostos.....	12
Figura 2.6. Pontos característicos ou minúcias – cristas finais e cristas bifurcadas na imagem de impressão digital	13
Figura 2.7. Esquemas de Classificação do FBI	14
Figura 2.8. Componentes das minúcias	15
Figura 3.1. Detecção de Bordas pelo Detector de “ <i>Canny</i> ”	18
Figura 3.2. Verificação dos 8 <i>pixels</i> vizinhos de <i>P</i> na direção anti-horária	20
Figura 3.3. Exemplos de crista final e crista bifurcada	20
Figura 3.4. Componentes das minúcias	21
Figura 3.5. Direção da Crista Local	22
Figura 3.6. Exemplo da Direção da Crista Local do Tipo Final.....	22
Figura 3.7. Exemplo da Direção da Crista Local do Tipo Bifurcada	23
Figura 4.1. As minúcias representadas por círculos juntamente com o centróide sobrepostos à imagem da impressão digital	27
Figura 4.2. Dois grafos distintos para mesma impressão digital.....	27

Figura 4.3. Os principais passos da abordagem. A figura mostra os resultados intermediários produzidos durante a classificação de uma Impressão digital.....	29
Figura 4.4. Sistema de Verificação de Impressão Digital <i>on-line</i>	30
Figura 4.5. Imagens de Impressão Digital de baixa qualidade	31
Figura 5.1. Diagrama da Metodologia Proposta	33
Figura 5.2. Resultado do Pré-processamento da Imagem	34
Figura 5.3. Extração de Minúcias Finais e Bifurcadas.....	34
Figura 5.4. Minúcias Finais, Bifurcadas e Centro de Massa.....	35
Figura 5.5. Geração do Vetor Distância ordenado do primeiro ao quarto quadrante	36
Figura 5.6. Geração do Vetor Distância das coordenadas rotacionadas de um ângulo θ	37
Figura 5.7. Minúcias com a origem do sistema de coordenadas deslocada para o Centro de Massa.....	38
Figura 5.8. Minúcias com valor da Distância Euclidiana ao Centro de Massa superior a 10 e inferior a 80	39
Figura 5.9. Vetor Distância A (original) e Vetor Distância B (rotacionada de um ângulo θ)	42
Figura 5.10. Comparação do Vetor de Alinhamento da primeira combinação do Vetor Distância A com o Vetor Distância B, gerando novo vetor e novo vetor ordenado	42
Figura 5.11. Validação dos três elementos do novo vetor que não ultrapassaram o valor de limiar 2.5 da primeira combinação do Vetor Distância A	43
Figura 5.12. Comparação do Vetor de Alinhamento da segunda combinação do Vetor Distância A com o Vetor Distância B, gerando novo vetor e novo vetor ordenado	43
Figura 5.13. Validação dos três elementos do novo vetor que não ultrapassaram o valor de limiar 2.5 da segunda combinação do Vetor Distância A	44

Figura 5.14. Comparação do Vetor de Alinhamento da terceira combinação do Vetor Distância A com o Vetor Distância B, gerando novo vetor e novo vetor ordenado	45
Figura 5.15. Validação dos quatros elementos do novo vetor que não ultrapassaram o valor de limiar 2.5 da terceira combinação do Vetor Distância A	45
Figura 5.16. Vetor Espúrio de Registro e Vetor Espúrio de Autenticação	46
Figura 6.1. Imagem Original da Impressão Digital e a mesma Imagem Rotacionada de 180°	50
Figura 6.2. Imagens Pré-processadas de Impressão Digital	51
Figura 6.3. Extração de Minúcias Finais e Bifurcadas da Imagem Original.....	51
Figura 6.4. Extração de Minúcias Finais e Bifurcadas da Imagem Rotacionada de 180°	52
Figura 6.5. Minúcias Finais e Bifurcadas com a origem deslocada para o Centro de Massa da Imagem Original.....	53
Figura 6.6. Minúcias Finais e Bifurcadas com a origem deslocada para o Centro de Massa da Imagem Rotacionada de 180°	53
Figura 6.7. Gráfico <i>Recall x Precision</i> do Método Média	61
Figura 6.8. Gráfico <i>Recall x Precision</i> do Método Normal	61
Figura 6.9. Caso relevante da 1ª a 5ª imagem mais próxima, buscando a imagem da pessoa número 1, quarta captura, rotacionada em 180°	62
Figura 6.10. Caso irrelevante na 2ª imagem mais próxima, buscando a imagem da pessoa número 10, quarta captura, rotacionada em 180°	63
Figura 6.11. Imagens que ocasionaram o insucesso algoritmo.....	63
Figura A.1. Exemplos da Direção de Crista Local Bifurcada onde $\theta=0^\circ$	71
Figura A.2. Exemplos da Direção de Crista Local Bifurcada onde $\theta=45^\circ$	72
Figura A.3. Exemplos da Direção de Crista Local Bifurcada onde $\theta=90^\circ$	73
Figura A.4. Exemplos da Direção de Crista Local Bifurcada onde $\theta=135^\circ$	73

Figura A.5. Exemplos da Direção de Crista Local Bifurcada onde $\theta=180^\circ$	74
Figura A.6. Exemplos da Direção de Crista Local Bifurcada onde $\theta=225^\circ$	75
Figura A.7. Exemplos da Direção de Crista Local Bifurcada onde $\theta=270^\circ$	76
Figura A.8. Exemplos da Direção de Crista Local Bifurcada onde $\theta=315^\circ$	76

Índice de Tabelas

Tabela 1.1. Comparação entre características biométricas	05
Tabela 3.1. Propriedades de Número de Cruzamentos	20
Tabela 6.1. Vetor Distância de Registro e Vetor Distância de Autenticação das Minúcias Finais.....	54
Tabela 6.2. Vetor Distância de Registro e Vetor Distância de Autenticação ordenado a partir do ponto de partida das Minúcias Finais	55
Tabela 6.3. Vetor Espúrio de Registro e Vetor Espúrio de Autenticação das Minúcias Finais.....	56
Tabela 6.4. Pior caso de Vetor Distância de Registro e Vetor Distância de Autenticação das Minúcias Finais.....	57
Tabela 6.5. Vetor Espúrio de Registro e Vetor Espúrio de Autenticação das Minúcias Finais com minúcias espúrias em comum.....	58
Tabela 6.6. Vetor Distância de Registro e Vetor Distância de Autenticação das Minúcias Finais com minúcias espúrias em comum re-incluídas	58
Tabela 6.7. Vetor Espúrio de Registro e Vetor Espúrio de Autenticação das Minúcias Finais após a re-inclusão das minúcias espúrias em comum.....	59

Lista de Abreviaturas

PIN	Número de Identificação Pessoal
IAFIS	Serviço de Identificação de Impressão Digital Automatizado
NIST	<i>American National Standards Institute</i>
CN	Número de Cruzamento
SAM	<i>Spatial Access Methods</i>
MAM	Métodos de Acesso Métricos
FBI	<i>Federal Bureau of Investigation</i> dos Estados Unidos

Resumo

O objetivo deste trabalho é apresentar algoritmos aplicados para extração de características invariantes à rotação em imagens de impressões digitais. No pré-processamento da imagem utiliza-se Detecção de Bordas pelo Detector de “*Canny*” tendo como resultado uma imagem binarizada e afinada. Na extração das minúcias a metodologia adotada é o Número de Cruzamentos (CN), que extrai os aspectos locais, tais como, as minúcias fim de linha e bifurcações. A direção das cristas locais não é utilizada porque nas imagens rotacionadas a condição de permanência das propriedades biométricas não são satisfeitas. A comparação das impressões digitais utiliza os vetores gerados pela extração de minúcias considerando a posição (x,y) da minúcia armazenada em um vetor por tipo de minúcia (um vetor para crista final e outro vetor para crista bifurcada) e calculando a Distância Euclidiana dessa posição (x,y) ao Centro de Massa da distribuição de minúcias para cada tipo de minúcia. Assim, as duas imagens são similares quando a Distância Euclidiana entre os vetores de cada imagem e por tipo de minúcia forem mínimas. São discutidas as limitações de outros trabalhos existentes envolvendo rotação, translação e distorção da imagem de impressão digital, mostrando que os poucos trabalhos que tratam o problema possuem resultados não satisfatórios. Os maiores problemas ocorridos foram a extração de minúcias espúrias, mas foram resolvidos com os métodos sugeridos por Dixon (1979), tendo resultados satisfatórios em duas metodologias. No método Média, a precisão para encontrar uma imagem foi de 100%, duas imagens 97,32%, três imagens 92,35%, quatro imagens 86,41% e cinco imagens 71,86%. E no método Normal, a precisão para encontrar uma imagem foi de 100%, duas imagens 99,20%, três imagens 96,95%, quatro imagens 94,00% e cinco imagens 76,43%.

Palavras-Chave: Impressão Digital, Invariância com relação à rotação, Reconhecimento de Padrões.

Abstract

The objective of this research is to present algorithms that can be applied in fingerprints images in order to extract certain features, which are invariant to an likely rotation in the given image. In the preprocessing stage, the "Canny" Border Detector is used, resulting in a binary, fine tuned image. For the minutiae extraction, the Crossing Number method is used, which extracts local aspects such as minutiae endings and bifurcations. The direction of local ridges is ignored because, in rotated images, the permanence conditions of the biometric attributes are not fulfilled. The process of matching fingerprints uses two arrays (one for ridge endings and the other for bifurcations), which are generated by the extraction of the minutiae, considering the (x,y) coordinate of the given minutiae stored in the arrays, and calculating its Euclidian Distance relating to the Center of Mass of the minutiae distribution, for each of its types (ending or bifurcation). Thus, both images are similar when the Euclidian Distance between the arrays of each image, distinct by the type of each minutiae, is minimal. The limitations of other pieces of research works concerning fingerprint image rotation, translation and distortion are discussed, indicating that the only few ones that deal with these kinds of problems give unsatisfactory results.

Keywords: Fingerprint Recognition, Rotation Invariance, Pattern Recognition.

Capítulo 1

Introdução

1.1. Considerações Iniciais

Métodos tradicionais para validação de identidade normalmente fazem uso de uma informação conhecida como senha e identificador de usuário, ou cartões de identificação e PIN (*Personal Identification Number*) (RATHA *et al*, 2001). O grande problema desses recursos é que podem ser esquecidos, roubados, perdidos, copiados, armazenados de maneira insegura e até utilizados por uma pessoa que não tenha autorização. Dessa forma, é justificável o uso da biometria, por ser um método de autorização e verificação muito mais exato, confiável e que possibilita identificar as pessoas por intermédio de suas características que definem sua individualidade sem que elas estejam carregando algum objeto ou memorizando senhas.

O reconhecimento de impressões digitais tem se destacado no contexto do reconhecimento de padrões, tendo como foco o desenvolvimento de sistemas automáticos que visam facilitar o reconhecimento de indivíduos utilizando suas características físicas. Com a evolução da tecnologia na era da informação, a sociedade vem se envolvendo com os sistemas eletrônicos e se sujeitando às fraudes. Daí a necessidade de desenvolver sistemas que facilitem a vida das pessoas e dificultem as fraudes, que estão aumentando e evoluindo. Na área forense o reconhecimento de impressão digital pode auxiliar a identificação de cadáveres, a investigação criminal, a identificação terrorista, a determinação de parentesco, o extravio de crianças, etc. Na área governamental pode ser usado no cartão de identificação nacional, na carteira de motorista, no seguro social, no controle de passaportes, no controle de fronteira, etc. Na área comercial o reconhecimento de impressão digital pode ser aplicado na senha da rede de computadores, na segurança de dados eletrônicos, no comércio eletrônico, no acesso à internet, no cartão de crédito, no telefone celular, no ensino à distância, etc., dentre muitas outras aplicações.

1.2. Biometria

Biometria é um conjunto de métodos automatizados, com base em características comportamentais (gestos, voz, modo de andar, escrita manual e assinatura) ou fisiológicas (impressão digital, face, geometria da mão, íris, veias da retina, voz e orelha). Essas características são chamadas de identificadores biométricos (Figura 1.1).

Existem casos em que os identificadores biométricos são uma combinação de características comportamentais e fisiológicas e eles não devem ser exclusivamente classificados como uma ou outra. Por exemplo, as impressões digitais podem ser do tipo fisiológico, mas o modo com que o usuário expõe o dedo no leitor da impressão digital, depende do comportamento dessa pessoa. Assim, a classificação é a combinação das características fisiológicas e comportamentais. (MALTONI *et al*, 2003)

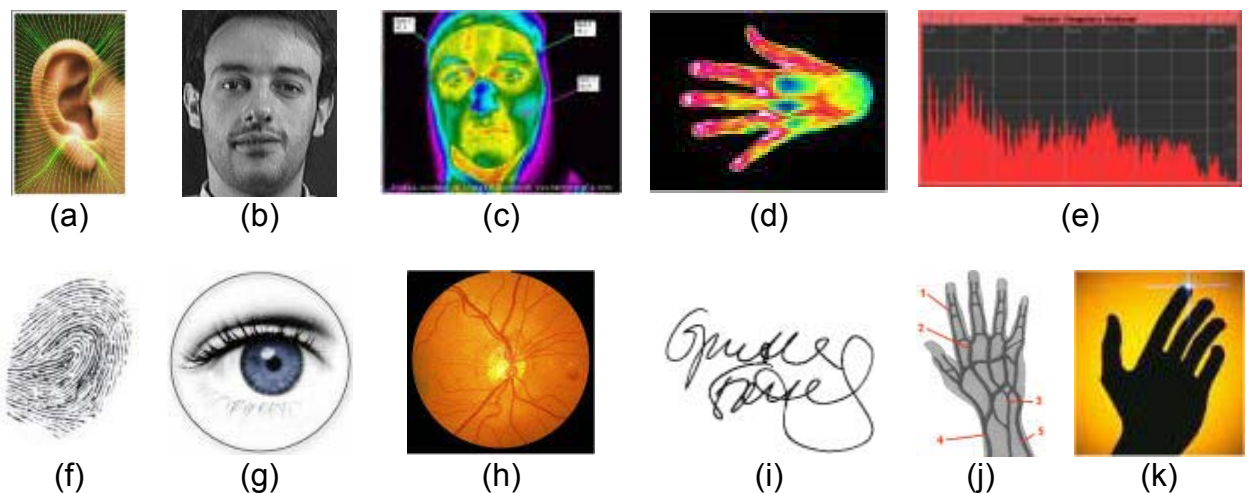


Figura 1.1: Alguns identificadores biométricos que são usados em várias aplicações: a) orelha, b) face, c) termograma facial, d) termograma da mão, e) voz, f) impressão digital, g) íris, h) retina, i) assinatura, j) veias da mão, k) geometria da mão

Uma propriedade biométrica é uma propriedade intrínseca de uma pessoa, é difícil fraudar e quase impossível compartilhar. Além disso, uma propriedade biométrica de uma pessoa pode ser perdida somente em acidentes sérios (RATHA *et al*, 2001). Essas propriedades podem ser consideradas como biométricas apenas se satisfizerem as seguintes condições:

- Universalidade: significa que toda pessoa deve possuir o identificador biométrico.
- Unicidade: indica que não pode existir duas pessoas que possuam o identificador biométrico com as mesmas características.
- Permanência: significa que o identificador biométrico deve ser suficientemente invariável por um período de tempo.
- Quantitatividade: significa que o identificador biométrico pode ser medido quantitativamente. (MALTONI *et al*, 2003) (JAIN *et al*, 2002)

O sistema biométrico de **identificação** determina a identidade de uma pessoa sem ela ter declarado um número de identificação ou o nome. Isso conduz a uma comparação “um para muitos”, pois procura num banco de dados inteiro uma identidade compatível. A pergunta utilizada nesse sistema será: “Quem sou eu?”

O sistema biométrico de **autenticação** autentica a identidade declarada de uma pessoa comparando a característica biométrica capturada com seu próprio “template” biométrico armazenado num sistema. Isso conduz a uma comparação “um para um” para determinar se a identidade declarada da pessoa é verdadeira. A pergunta utilizada nesse sistema será: “Eu sou quem declaro ser?”

A Figura 1.2 ilustra os diagramas dos sistemas de identificação e autenticação. O registro do usuário é comum nas duas tarefas (Figura 1.2.a). O módulo de registro é responsável por registrar as pessoas num banco de dados biométrico. Durante a fase de registro, as características da pessoa são primeiramente capturadas por um leitor biométrico para produzir a representação digital bruta das características. Um controle de qualidade é geralmente executado para assegurar que a amostra adquirida possa ser processada com confiança pelas fases sucessivas. Assim, para facilitar a procura, a representação digital bruta é geralmente processada mais adiante por um extrator de características para gerar uma representação expressiva, chamada de “template”. Dependendo da aplicação, o “template” pode ser armazenado na base de dados central do sistema biométrico ou ser gravado num cartão magnético pessoal.

A tarefa de autenticação (Figura 1.2.b) é responsável por verificar pessoas na hora do acesso. Durante a fase de operação, o nome do usuário ou o número de identificação pessoal (PIN) é informado por um teclado; o leitor biométrico captura as

características da pessoa para ser reconhecida e converte para um formato digital, que é mais adiante processada pelo extrator de características para produzir uma representação digital compacta. A representação resultante é fornecida ao comparador de características, que compara com o “template” de um único usuário, acessando a base de dados do sistema baseado no número de identificação pessoal (PIN).

Na tarefa de identificação (Figura 1.2.c), não é fornecido o número de identificação pessoal (PIN) e o sistema compara a representação da entrada biométrica com os “templates” de todos os usuários no banco de dados; a saída é a identidade de um usuário registrado ou uma mensagem de alerta tal como “usuário não identificado”.

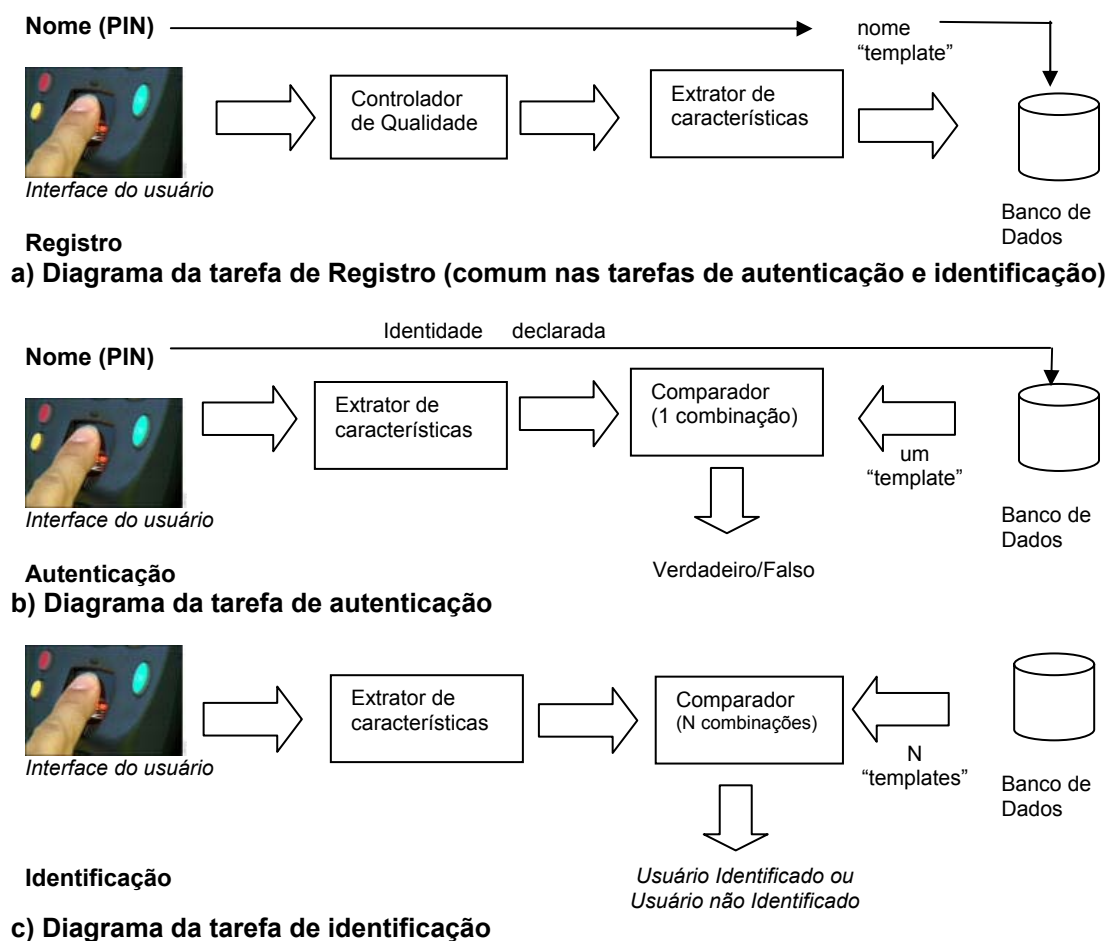


Figura 1.2: Diagramas das tarefas de registro, autenticação e de identificação (MALTONI et al, 2003)

O principal enfoque do estudo das tecnologias biométricas, no contexto geral, está voltado para o desenvolvimento de sistemas que permitam o reconhecimento e verificação de padrões, sendo de extrema importância o tempo de processamento, a segurança e a viabilidade das tecnologias biométricas.

A tabela 1.1 apresenta uma comparação entre diversas características biométricas utilizadas para reconhecimento e identificação de pessoas (BOLLE *et al*, 2001). Nessa comparação foram avaliados a facilidade de uso, a incidência de erros, a precisão, a aceitação dos usuários, o nível de segurança e a estabilidade da característica biométrica durante a vida. Dentre todos esses parâmetros de comparação, a impressão digital apresenta-se como uma das características mais adequadas para o uso em sistemas de reconhecimento.

Tabela 1.1: Comparação entre características biométricas (LIU *apud* BOLLE, 2001)

Característica	Impressão Digital	Geometria da mão	Retina	Íris	Face	Assinatura	Voz
Facilidade de uso	Alta	Alta	Baixa	Média	Média	Alta	Alta
Incidência de erros	Umidade, sujeira, idade	Acidentes, idade	Óculos	Pouca Luz	Iluminação, idade, óculos, cabelo	Mudanças na assinatura	Ruído, resfriado, clima
Precisão	Alta	Alta	Muito alta	Muito alta	Alta	Alta	Alta
Aceitação	Média	Média	Média	Média	Média	Média	Média
Nível de Segurança	Alto	Médio	Alto	Muito alto	Médio	Médio	Médio
Estabilidade ao longo do tempo	Alta	Média	Alta	Alta	Média	Média	Média

A geometria da mão apresenta boas características obtidas do formato da mão. As características geralmente medidas após a captura da imagem são o comprimento e largura dos dedos e a largura da mão. O problema é que a geometria da mão pode variar com ganho ou perda de peso, com a idade, doenças e acidentes em geral (JAIN *et al*, 2000).

A varredura de retina é bastante precisa, mas requer que o usuário olhe para um ponto fixo através de um equipamento, que analisa a camada de vasos sanguíneos localizada na parte de trás do globo ocular. Para isso, utiliza-se uma fonte de luz de baixa intensidade que varre o padrão da retina. Essa varredura não é muito bem aceita, devido ao receio de danos ao olho quando em contato com o aparelho (JAIN *et al*, 2000).

A íris é a parte colorida do olho ao redor da pupila e permanece estável para a vida inteira (SANDERSON *et al*, 2000). Cada íris é única. Praticamente não existem duas íris iguais, mesmo nos gêmeos idênticos (monozigótico). A probabilidade de duas pessoas possuírem o mesmo padrão de íris é muito grande. Além disso, as íris dos olhos direito e esquerdo de uma mesma pessoa possuem padrões de textura diferentes (DAUGHMAN *apud* JAIN *et al*, 2000).

A característica biométrica da face pode ser obtida das características genéticas faciais, ou por análise global. As abordagens geométricas utilizadas são a localização e a determinação do formato dos atributos faciais (olhos, boca, sobrancelhas, nariz, queixo, etc.) e o relacionamento entre eles. A decomposição da face em um número de faces canônicas é uma metodologia de análise global que não leva em conta as características geométricas. O grande problema é relacionado a variações de luz, idade, óculos, cabelo, cor da pele, posição, etc... (CHELLAPPA *apud* JAIN *et al*, 2000) em ambas as abordagens.

A característica biométrica da assinatura permite analisar a maneira como um usuário assina seu nome, verificando o formato da escrita, a velocidade e a pressão da caneta sobre o papel. O principal problema é a variação na assinatura de um mesmo indivíduo (JAIN *et al*, 2000).

A voz é uma característica biométrica fisiológica e comportamental e seu reconhecimento lida com a extração de características do sinal de voz sendo independente da linguagem. Neste tipo de característica biométrica não é importante o que está sendo dito, mas sim como está sendo dito. Os erros são incidentes em casos de ruído, resfriado e clima (JAIN *et al*, 2000).

Este trabalho visa obter a viabilidade de se implantar na prática um sistema biométrico de Autenticação por Impressão Digital, considerando rotações em qualquer ângulo. Com o objetivo de colaborar com a evolução da tecnologia, facilitando o reconhecimento de indivíduos aumentando a segurança nas áreas forense, comercial e governamental.

1.3. Estrutura da Dissertação

No primeiro capítulo são definidos alguns conceitos e definições básicas sobre biometria, utilizados no decorrer deste trabalho. No capítulo dois são descritos a análise e a representação das impressões digitais, apresentando os aspectos globais, os pontos singulares (núcleos e deltas) e os aspectos locais; abordando também as minúcias ou Detalhes de Galton e ressaltando a comparação entre classificação contínua e classificação exclusiva. As técnicas destinadas à extração de características, destacando-se a importância do pré-processamento de imagens, são descritas no capítulo três. No capítulo quatro, são discutidas as limitações de outros trabalhos existentes envolvendo rotação, translação e distorção da imagem de impressão digital. A metodologia proposta neste trabalho é apresentada no capítulo cinco e os resultados obtidos, conclusões e perspectivas futuras são apresentadas no capítulo seis.

Capítulo 2

Impressão Digital

2.1. Introdução

As impressões digitais são formadas geralmente no sétimo mês de gestação e permanecem intactas por toda a vida, se não houver um ferimento ou corte profundo (JAIN *et al*, 2002). Esta propriedade faz com que a impressão digital seja um grande atrativo na identificação biométrica.

Quanto à aquisição das imagens de uma impressão digital, por muitos anos foi baseada em tinta e papel (SENIOR *et al*, 2002) utilizando-se dois métodos: o primeiro método e mais comum é quando o dedo coberto de tinta é rolado de um lado ao outro num pedaço de papel, de maneira que o desenho digital não apresente borrões ou manchas. Este método captura uma superfície maior do dedo, mas na prática, estas imagens apresentam problemas no ato de rolar o dedo, podendo causar distorções no resultado da imagem por excesso ou falta de tinta, afetando notavelmente a qualidade da imagem. O segundo método implica em apenas pressionar o dedo coberto de tinta sobre o papel, sem rolar o dedo. Este método causa menos distorções na imagem da impressão digital porque o esticamento é minimizado, mas menos superfície do dedo é capturada (ADHAMI *et al*, 2001). Entretanto, a qualidade da aquisição baseada em tinta e papel não corresponde a um padrão aceitável, e a verificação automática da impressão digital torna-se uma tarefa extremamente difícil e imprópria. É possível observar a diferença na qualidade das imagens entre os métodos de aquisição, para a mesma imagem coletada em papel e utilizando um leitor biométrico (Figura 2.1). O leitor biométrico utiliza um sistema eletrônico de geração de dados, e por esta razão é muito mais eficiente. O leitor biométrico transforma os aspectos físicos extraídos em um “template”, ou seja, em um conjunto de características (JAIN *et al*, 2000).



Figura 2.1: Métodos de Aquisição da Imagem de Impressão Digital
(à esquerda impressão digital adquirida por tinta e papel e
à direita com sensor (SENIOR *et al*, 2002))

A análise manual da impressão digital é uma tarefa tediosa, demorada e com alto custo que não se enquadra às exigências de desempenho das aplicações novas (JAIN *et al*, 1997), onde os aspectos para comparação são extremamente pequenos necessitando auxílio de lentes de aumento para obter um melhor exame da marca da impressão digital (RATHA *et al*, 1995).

Outro ponto a ser considerado é o tamanho da base de dados, o que pode fazer a análise e comparação manual levar dias em alguns casos. Estes problemas podem ser facilmente superados pela automação do processo de verificação da impressão digital visando otimizar o processamento.

2.2. Conceitos Biométricos para Impressão Digital

Todas as pessoas possuem uma impressão digital que é única e permanente, o que demonstra os critérios de universalidade, unicidade e permanência. As digitais apresentam um conjunto de características pessoais que pode ser medido, satisfazendo o critério quantitativo (ELECCION, 1973). Assim, as impressões digitais podem ser usadas como características biométricas.

Entretanto, na prática dos sistemas biométricos, existem outros assuntos que devem ser considerados, incluindo:

- Desempenho: refere-se aos recursos operacionais ou fatores ambientais necessários para conseguir um reconhecimento preciso, rápido e robusto;
- Acessibilidade: indica o nível que as pessoas estão dispostas a aceitar um identificador biométrico particular em suas vidas diárias;

- Nível de Fraude: pondera a facilidade em enganar o sistema por métodos fraudulentos.

Maltoni (2003) comparou as tecnologias biométricas e mostrou que a impressão digital possui alto desempenho, média acessibilidade e nível médio de fraudes.

2.3. Análise e Representação das Impressões Digitais

A impressão digital é uma reprodução da epiderme da ponta do dedo, produzida quando o dedo é pressionado contra uma superfície lisa. A característica estrutural mais evidente é um padrão de vales e cristas intercaladas; na imagem de impressão digital, as cristas são pretas enquanto que os vales são brancos (Figura 2.2).

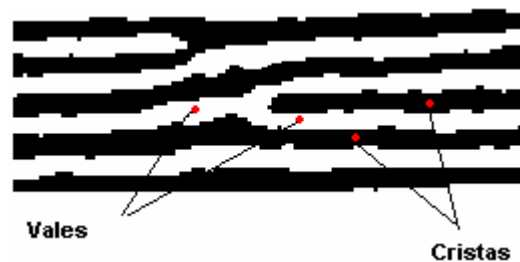


Figura 2.2: Cristas e Vales.

Quando são analisados as cristas e os vales no *aspecto global*, o padrão da impressão digital exibe uma ou mais regiões onde as cristas assumem diferentes formas (caracterizando curvatura alta, terminações freqüentes, etc). Essas regiões são chamadas de singularidades ou regiões singulares.

As impressões digitais possuem *aspectos locais* como cristas finais, cristas bifurcadas, cristas curtas, cruzamentos, esporas e ilhas, chamados de minúcias. Francis Galton foi o primeiro a classificar as minúcias e observar que elas permanecem inalteradas durante toda a vida do indivíduo. As minúcias são chamadas de “Detalhes de Galton” em sua homenagem.

2.3.1. Pontos Singulares (Núcleos e Deltas)

A área padrão das impressões digitais é a área que contém os pontos singulares e as cristas primárias para classificação (MOAYER *apud* CHANG *et al*, 2002).

O desenho digital apresenta uma área útil onde é possível observar a área padrão e tipos de linhas (Figura 2.3).

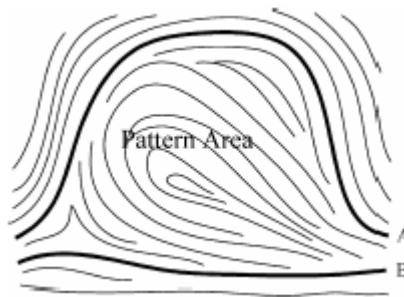


Figura 2.3: Área padrão (*Pattern Area*) e tipos de linhas (*A e B*) (CHANG *et al*, 2002).

A área padrão é a parte principal da impressão do dedo e consiste das cristas e todos os seus aspectos.

Tipos de linhas são as cristas que delimitam a área padrão. Elas devem iniciar paralelamente e divergir em direção oposta quando elas alcançam a área padrão. A existência de tipos de linhas é muito importante para classificação da impressão digital. Se não for reconhecida, a imagem da impressão digital provavelmente é uma imagem parcial que conduzirá à desclassificação.

Os pontos singulares em impressões digitais são conhecidos como núcleos e deltas, que são usados para classificar os padrões de impressões digitais (Figura 2.4).



Figura 2.4: Singularidades, geralmente classificadas em Núcleo (*core*) e Delta. (MAIO *et al*, 1996)

O núcleo é um ponto localizado na área central da impressão digital.

Delta é um ângulo ou triângulo formado pelas cristas papilares e pode ser constituído por dois processos: pela bifurcação de uma linha simples ou pela brusca divergência de duas linhas paralelas. O delta desempenha papel importante na impressão digital porque determina a sua classe.

2.3.2. Minúcias ou “Detalhes de Galton”

As minúcias podem ser classificadas dentro de duas categorias: aspectos básicos e aspectos compostos. Os aspectos básicos são: cristas finais e cristas bifurcadas. Os aspectos compostos são: ilhas, cristas curtas, espora, cruzamento e são construídos a partir dos aspectos básicos. Ambos os aspectos estão detalhados na Figura 2.5.

Galton (1982) definiu um conjunto de características para identificação da impressão digital. Entretanto, muitas dessas características não são usadas nos sistemas de identificação da impressão digital. Ao invés disso, as minúcias são restritas a somente dois tipos, cristas finais e bifurcadas. Cristas finais são os pontos onde a crista termina, e as bifurcadas são onde a crista se divide de um único caminho para dois formando um entroncamento Y.

O padrão de coordenada dessas minúcias é único para um dado indivíduo; assim, pode servir como identificação.

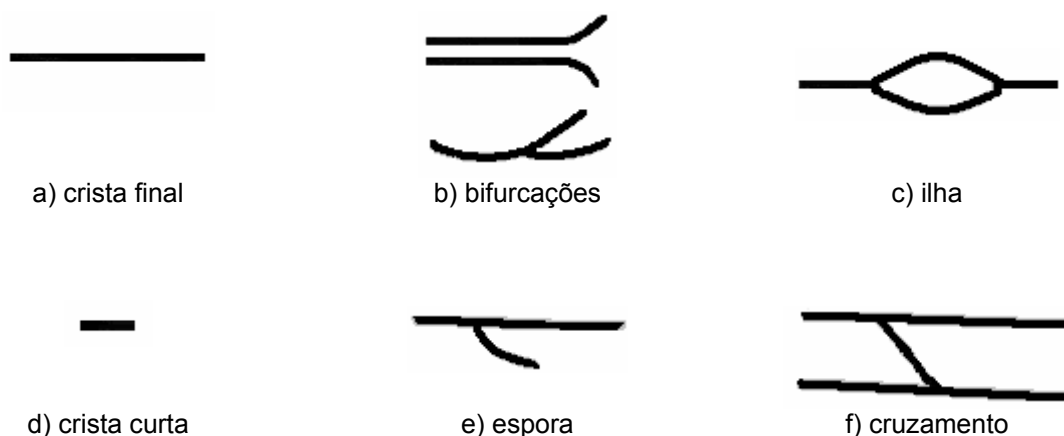


Figura 2.5: (a) e (b) são os aspectos básicos e (c) a (f) aspectos compostos

As minúcias são, então, acidentes que se encontram nas cristas papilares como, por exemplo, linhas que terminam abruptamente ou se bifurcam, e tem a finalidade de estabelecer a unicidade das impressões digitais (Figura 2.6).

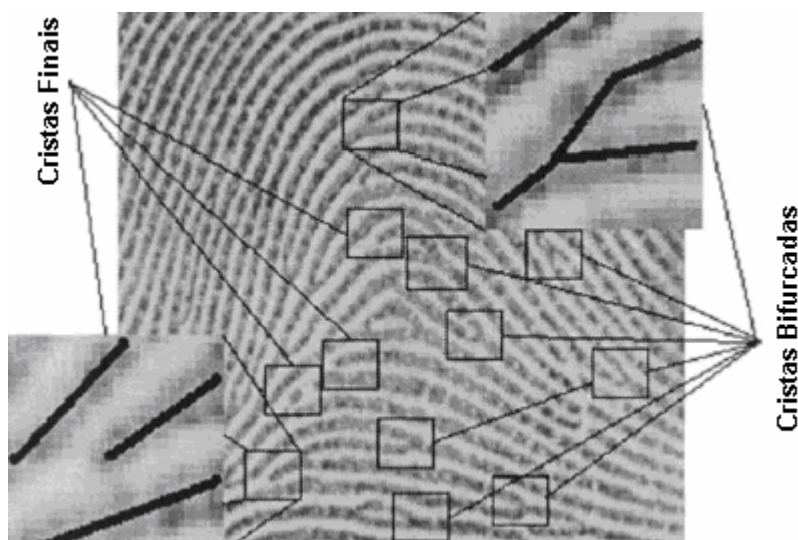


Figura 2.6: Pontos característicos ou minúcias - cristas finais e cristas bifurcadas na imagem de impressão digital (GREENBERG, 2002)

O *NIST* (*American National Standards Institute*) propôs quatro grupos principais de minúcias: cristas finais, bifurcações, cruzamentos e pontos isolados. Porém as minúcias consideradas mais importantes são as cristas finais e bifurcações, pois ocorrem freqüentemente nas imagens de impressões digitais (FARINA *et al*, 1999), (HRECHAK *et al*, 1990).

Para a verificação de impressões digitais é, então, necessário:

- Que haja coincidência de doze pontos característicos no mínimo (ELECCION, 1973).
- Que não exista nenhuma discordância entre estes pontos, ou seja, eles devem ser idênticos e ter a mesma localização.

2.4. Classificação Exclusiva X Classificação Contínua

O reconhecimento da impressão digital é uma tarefa básica de um Serviço de Identificação de Impressão Digital Automatizado (IAFIS). Quase todos IAFIS usam a *classificação exclusiva*, isto é, as impressões digitais são divididas em algumas

classes pré-definidas de acordo com suas macro-características. Os primeiros estudos científicos na classificação da impressão digital, feitos por Galton (GALTON *apud* CAPPELLI *et al*, 1999), dividiu as impressões digitais em três classes principais. Mais tarde, Henry (HENRY *apud* CAPPELLI *et al*, 1999) refinou a classificação de Galton para aumentar o número de classes. O esquema adotado pelo FBI (*Federal Bureau of Investigation* dos Estados Unidos) define três classes principais, e cada uma pode ser dividida em duas ou mais subclasses, perfazendo o total de oito: *Plain Arch*, *Tented Arch*, *Radial Loop*, *Ulnar Loop*, *Plain Whorl*, *Central Pocket*, *Double Loop* and *Accidental Whorl* (Figura 2.7) (CAPPELLI *et al*, 1999).

O desempenho baseado na *classificação exclusiva* de IAFIS, depende fortemente do número de classes e da distribuição nas impressões digitais; infelizmente, o número dessas classes é frequentemente pequeno, as impressões digitais são distribuídas não-uniformemente (nos esquemas de classificação mais famosos, aproximadamente 90% das impressões digitais pertencem a somente três classes), e existem impressões digitais “ambíguas”, cuja classificação exclusiva não pode ser indicada confiantemente nem mesmo pelos peritos humanos. (CAPPELLI *et al*, 1999).



Figura 2.7: Cada impressão digital na figura pertence a subclasses diferentes no esquema de classificação do FBI (*Federal Bureau of Investigation* dos Estados Unidos): da esquerda para direita e de cima para baixo: Plain Arch, Tented Arch, Radial Loop, Ulnar Loop, Plain Whorl, Central Pocket, Double Loop, and Accidental Whorl (CAPPELLI *et al*, 1999)

As representações da impressão digital baseada na *classificação contínua* são referentes à imagem inteira, em cristas do dedo, em poros das cristas, ou nas características salientes derivadas das cristas. Tipicamente, as representações baseadas em minúcias contam com as posições das minúcias e nos sentidos das cristas na posição das minúcias (Figura 2.8) (JAIN *et al*, 2000b).

Lumini (1997) avaliou a capacidade para recuperar impressão digital latente em base de dados grandes. Comparou duas metodologias diferentes de classificação da impressão digital, utilizando a classificação exclusiva e contínua. Os resultados obtidos foram melhores nas duas metodologias, através da *classificação contínua*.

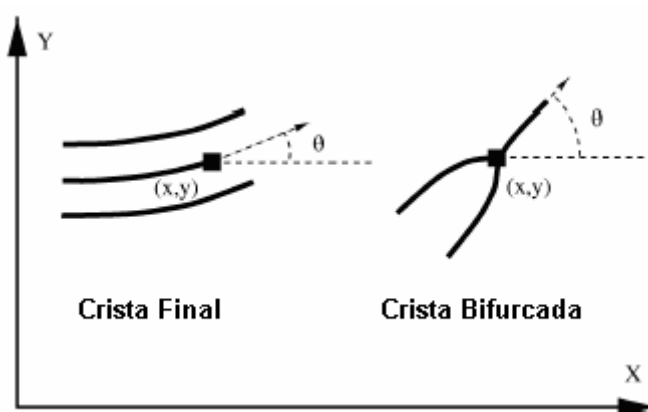


Figura 2.8: Componentes das minúcias (HONG *et al*, 1998b)

2.5. Considerações Finais

As impressões digitais de uma pessoa têm sido usadas desde a antiguidade como identificador biométrico confiável.

As metodologias de classificação exclusiva ou contínua têm sido largamente empregadas na literatura especializada com ligeira vantagem para a segunda.

No entanto, os métodos que consideram a rotação da imagem adquirida são raros, sendo que, a maior parte justifica sua validade dentro de pequenas variações angulares.

Este nosso trabalho visa verificar impressões digitais que sofreram rotação, utilizando algoritmos pela metodologia de classificação contínua, ou seja, extraíndo

as características derivadas das cristas finais e bifurcadas. A extração dessas características será realizada na imagem de impressão digital da mesma pessoa, na posição original e rotacionada em vários ângulos. Deverá então, ser comparada em um banco de imagens buscando pela imagem mais similar considerando a rotação.

Capítulo 3

Extração de Características

3.1. Introdução

Para que o processo de comparação de impressões digitais seja considerado confiável, é necessária a execução de etapas preliminares para o tratamento das imagens. Essas etapas iniciais podem envolver técnicas como binarização, afinamento, remoção de ruídos, filtros, detecção de bordas, entre outras.

Segundo Lindley (1991), as técnicas de processamento de imagens em geral são aplicadas quando se verificam as seguintes necessidades:

- Alguns aspectos da imagem precisam ser melhorados devido à presença de ruídos.
- Elementos da imagem precisam ser caracterizados, classificados, comparados ou medidos, como por exemplo, a extração de características em imagens de impressões digitais que possibilitem realizar a classificação e a verificação;
- É necessário combinar ou reorganizar determinadas regiões das imagens.

O objetivo final é a comparação de imagens a serem identificadas ou reconhecidas. Para isso são necessárias a classificação e a caracterização de elementos contidos na imagem. E para que estas fases preliminares sejam executadas com sucesso, é razoável que se disponha de uma imagem sem ruído (LINDLEY, 1991).

3.2. Pré-processamento

As imagens de impressões digitais raramente possuem uma boa qualidade, geralmente são irregulares com elementos de ruídos, devido a muitos fatores, incluindo variações nas condições da pele, marcas de nascimento, cortes, uso de cremes e também devido aos dispositivos de aquisição. Esses fatores influenciam

significativamente a qualidade das imagens, ou seja, nem sempre as estruturas das minúcias são bem definidas, portanto não podem ser detectadas ou podem resultar um número significativo de minúcias falsas. Assim, o primeiro passo a ser realizado como procedimento geral é a aplicação do limiar ou binarização (*threshold*), ou seja, tendo como entrada uma imagem de impressão digital em tons de cinza é necessário transformá-la em uma imagem binária. O segundo passo trata-se da aplicação do afinamento, ou seja, as cristas das impressões digitais passam por um processo de emagrecimento (JAIN *et al*, 1997). Essas duas técnicas de pré-processamento devem ser efetuadas antes da extração das minúcias para obter uma estimativa mais confiável de posições das minúcias detalhadas como cristas finais e cristas bifurcadas.

3.2.1. Detecção de Bordas pelo Detector de “Canny” (THAI, 2003)

Este método encontra as bordas procurando máximos locais do gradiente da imagem. O gradiente é calculado usando a derivada do Filtro Gaussiano. O método usa dois “*thresholds*” (limiar) para detectar bordas fortes e fracas, e considera as bordas fracas que são conectadas às bordas fortes. A imagem de entrada para este método é uma imagem em níveis de cinza e a saída é uma imagem binarizada e afinada (Figura 3.1).



Imagem Original

Imagem Pré-Processada

Figura 3.1: Detecção de Bordas pelo Detector de “Canny”

3.3. Extração de Aspectos

Existem dois tipos de aspectos que podem ser extraídos de imagens de impressões digitais: aspectos globais (núcleos e deltas) e aspectos locais (cristas finais e cristas bifurcadas). Os pontos singulares, núcleos e deltas, podem ser detectados do bloco de direção, como por exemplo, o cálculo da imagem direcional, a suavização do mapa direcional e o cálculo do índice de Poincaré como proposto no sistema de Henry (COSTA, 2001).

Para localizar as cristas finais e cristas bifurcadas há necessidade de seguir as linhas do esqueleto na imagem. Neste conjunto de *pixels* são encontradas as minúcias, usadas para a verificação. O conceito de Número de Cruzamentos (*Crossing Number* - CN) (THAI, 2003) é muito usado para detecção destes aspectos ou minúcias e determinam as propriedades de um *pixel* simplesmente contando o número de transições preto e branco existentes na vizinhança de 8 do *pixel* que está sendo processado.

3.4. Número de Cruzamentos (*Crossing Number* - CN) (THAI, 2003)

O processo de extração de minúcias permite sintetizar propriedades inerentes da imagem, que serão utilizadas no processo de armazenagem e recuperação baseada em seu conteúdo.

Um dos métodos mais utilizados para extração das minúcias é o Número de Cruzamentos (CN) (AMENGUAL *apud* THAI, 2003). Esse método extrai as minúcias fazendo uma varredura na vizinhança de 8 *pixels* usando uma janela 3x3. O valor de CN é definido como a metade da soma das diferenças entre os pares de *pixels* adjacentes na vizinhança de 8.

O CN para o *pixel* P é dado pela equação (3.1):

$$CN = 0.5 \sum_{i=1}^8 |P_i - P_{i+1}|, \quad P_9 = P_1 \quad (3.1)$$

Onde P_i é o valor do *pixel* na vizinhança de P . Para um *pixel* P , os 8 *pixels* vizinhos são examinados na direção anti-horária como na figura 3.2:

P_4	P_3	P_2
P_5	P	P_1
P_6	P_7	P_8

Figura 3.2: Verificação dos 8 *pixels* vizinhos de P na direção anti-horária

Como ilustra a Figura 3.3, o *pixel* com CN igual a 1 corresponde à crista final, e com CN igual a 3 corresponde à crista bifurcada. Para cada ponto de minúcia extraída, a seguinte informação é armazenada:

- Coordenadas x e y ,
- Direção da crista local (θ), e;
- Tipo de minúcia (crista final ou bifurcada = CN)

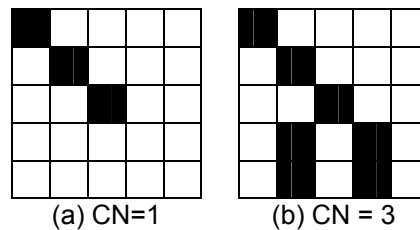


Figura 3.3: Exemplos de crista final e crista bifurcada

A Tabela 3.1 mostra as propriedades do CN, onde o *pixel* pode ser classificado como ponto isolado, crista final, linha contínua, crista bifurcada ou cruzamento de pontos.

Tabela 3.1: Propriedades do Número de Cruzamentos (THAI, 2003)

Número de Cruzamentos (propriedades)	Minúcias
0	Ponto Isolado
1	Final
2	Linha contínua
3	Bifurcação
4	Cruzamento de pontos

Cristas finais e cristas bifurcadas são os aspectos estruturais mais comuns extraídos de imagens de impressões digitais em nível cinza. Cada um destes dois tipos de minúcias tem normalmente três atributos: a coordenada x , a coordenada y e a direção da crista local (θ) (Figura 3.4) (HONG et al, 1998), (RATHA et al, 1996).

3.5. Direção da Crista Local

O FBI considera somente as terminações e bifurcações como o modelo de coordenadas de minúcia: cada minúcia é denotada pela posição das coordenadas x e y e o ângulo entre a tangente para as cristas na posição da minúcia e o eixo horizontal (Figura 3.4) (MALTONI et al, 2003).

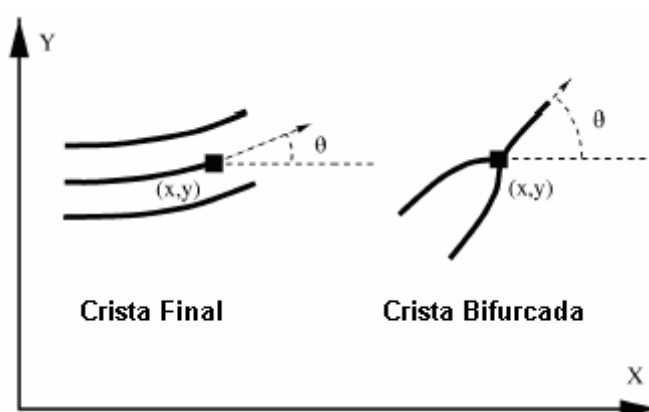


Figura 3.4: Componentes das minúcias (HONG et al, 1998b)

3.5.1. Direção da Crista Local do Tipo Final

A direção da crista local (θ) é encontrada após o cálculo de CN. Se a crista que está sendo processada for do tipo final, ou seja, CN igual a 1, o método apenas procura o próximo *pixel* preto, fazendo uma varredura na vizinhança de 8 *pixels* usando uma janela 3x3, os 8 *pixels* vizinhos são examinados na direção anti-horária utilizando os ângulos determinados na Figura 3.5.

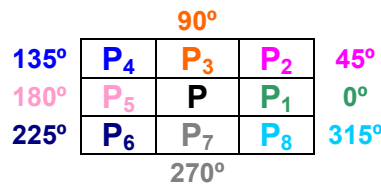


Figura 3.5: Direção da Crista Local

A direção da crista local é formada pela reta paralela ao eixo x e a tangente à curva no *pixel* P . A Figura 3.6 ilustra a crista do tipo final que passa pelo *pixel* P que está sendo processado e pelo *pixel* P_6 . Portanto, a direção da crista local neste exemplo será igual a 225°.

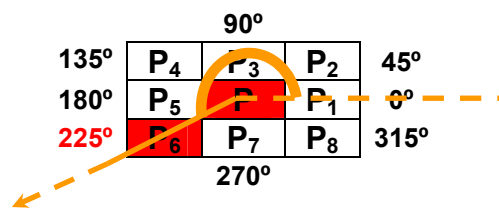


Figura 3.6: Exemplo da Direção da Crista Local do Tipo Final

3.5.2. Direção da Crista Local do Tipo Bifurcada

Se a crista que está sendo processada for do tipo bifurcada, ou seja, CN igual a 3, o método apenas procura o próximo *pixel* preto, fazendo uma varredura na vizinhança de 8 *pixels* usando uma janela 3x3. Os 8 *pixels* vizinhos são examinados na direção anti-horária utilizando os ângulos determinados na Figura 3.5.

A direção da crista local do tipo bifurcada é formada pela reta paralela ao eixo x e a tangente à curva no *pixel* P . A Figura 3.7 ilustra a crista do tipo bifurcada que passa pelo *pixel* P que está sendo processado, pelo *pixel* P_3 e bifurca entre os pixels P_6 e P_8 . Portanto, a direção da crista local para este exemplo será igual a 90°.

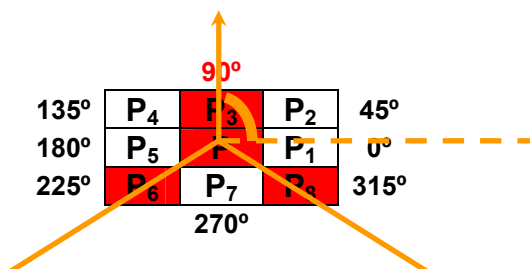


Figura 3.7: Exemplo da Direção da Crista Local do Tipo Bifurcada

As Regras para a Direção da Crista Local Bifurcada estão descritas no Apêndice A.

O maior problema relativo ao cálculo da direção das cristas é referente à rotação das imagens de impressões digitais, fato este possível de acontecer durante o processo de aquisição das imagens.

Sem que se conheça previamente o ângulo de rotação da imagem, fica difícil de se comparar, *a posteriori*, duas impressões da mesma pessoa que tenham sofrido rotação.

Neste nosso trabalho a direção das cristas locais não será utilizada como representação das minúcias, pois, pretende-se comparar uma imagem original com a mesma imagem rotacionada em vários ângulos. Portanto, a direção das cristas locais em imagens rotacionadas não satisfaz a condição de permanência das propriedades biométricas tornando a impressão digital um identificador inválido por ser variável em casos de imagens rotacionadas.

3.6. Comparação das Minúcias

Nos algoritmos que avaliam os aspectos globais analisando os pontos singulares, núcleos e deltas, uma dada impressão digital é comparada com todas as impressões digitais da sua respectiva classe. Dessa forma é possível diminuir o tempo de processamento e aumentar a precisão na comparação e reconhecimento. Mas, infelizmente, o número dessas classes é frequentemente pequeno, as impressões digitais são distribuídas não-uniformemente (nos esquemas de classificação mais famosos, aproximadamente 90% das impressões digitais pertencem a somente três classes), e existem impressões digitais “ambíguas”, cuja

sociedade exclusiva não pode ser indicada confiadamente nem mesmo pelos peritos humanos. (CAPPELLI *et al*, 1999).

3.6.1. Distância Euclidiana

Para os pixels p , q e z , com coordenadas (x,y) , (s,t) e (u,v) respectivamente, D é uma função distância ou métrica se

(a) $D(p,q) \geq 0$ ($D(p,q) = 0$ se e somente se $p=q$)

(b) $D(p,q) = D(q,p)$, e

(c) $D(p,z) \leq D(p,q) + D(q,z)$

A distância Euclidiana entre p e q é dada pela equação 3.2.:

$$D_e(p,q) = [(x - s)^2 + (y - t)^2]^{1/2} \quad (3.2)$$

Para essa medida de distância, os pixels tendo uma distância de (x,y) menor ou igual a algum valor r são os pontos contidos em um disco de raio r centrado em (x,y) . (GONZALEZ *et al*, 1992)

A Distância Euclidiana entre dois vetores (i e j) pode ser obtida diretamente da métrica de Minkowsky onde $p=2$, conforme mostra a equação 3.3.

$$d_{ij} = \left(|x_{i1} - x_{j1}|^p + |x_{i2} - x_{j2}|^p + \dots + |x_{ip} - x_{jp}|^p \right)^{1/p} \quad (3.3)$$

3.6.2. Centro de Massa

A definição física de Centro de Massa de um conjunto de partículas (m_1, m_2, m_3), cujas posições podem ser representadas pelos vetores posição

(r_1, r_2, r_3) respectivamente, em relação a um referencial inercial (posições relativas a um observador que seja ele próprio uma partícula ou sistema livre), é uma posição cujo vetor é definido pela equação 3.4.

$$r_{cm} = \frac{m_1 r_1 + m_2 r_2 + m_3 r_3}{m_1 + m_2 + m_3} = \frac{\sum m_i r_i}{M} \quad (3.4)$$

3.7. Considerações Finais

Os métodos de extração de características em imagens de impressões digitais preocupam-se com os aspectos globais e os aspectos locais.

Os aspectos globais sofrem do problema de desbalanceamento de classes, pois, a distribuição na população em geral é não homogênea e com problemas de ambigüidade.

Os aspectos locais podem classificar inequivocamente uma impressão digital, mas a característica de ângulo da minúcia é sensível à rotação da imagem.

Neste trabalho, o algoritmo proposto para comparação das impressões digitais, utiliza os vetores gerados pela extração das minúcias, e considera a posição (x, y) da minúcia armazenada em um vetor por tipo de minúcia (um vetor para crista final e outro vetor para crista bifurcada). Para que fique invariante com relação à rotação da impressão digital, gera-se um novo vetor (Vetor Distância), formado pela Distância Euclidiana dessa posição (x, y) ao Centro de Massa da distribuição espacial para cada tipo de minúcia. Dessa forma, duas imagens serão similares quando a Distância Euclidiana calculada entre os Vetores Distância de cada imagem e por tipo de minúcia forem mínimas.

Capítulo 4

Métodos para o reconhecimento que consideram a rotação de Impressão Digital

4.1. Introdução

Os sistemas de reconhecimento automático que utilizam a característica física, impressão digital, ainda não são totalmente seguros. Na ocorrência de problemas na captura das imagens de impressões digitais alguns pontos importantes, para o processo de comparação, podem ser perdidos e falsos positivos/negativos podem acontecer. No caso de imagens de impressões digitais rotacionadas pontos podem ser perdidos e outros podem surgir, durante a extração de minúcias, quando se aplica um ângulo de rotação à imagem afetando o processo de reconhecimento.

4.2. Algumas Técnicas Empregadas

O objetivo do trabalho de Tedesco e outros (2001) foi desenvolver um algoritmo utilizando técnicas de tratamento de imagem e de grafos para minimizar o problema causado pela rotação das imagens, na etapa da captura, que influenciam no confronto das impressões digitais. As técnicas de pré-processamento foram a binarização e o afinamento, alegando que com as imagens na forma binária é mais simples trabalhar e a técnica de afinamento define melhor as formas geométricas da impressão digital facilitando o reconhecimento das minúcias.

O objetivo do grafo foi construir uma estrutura eficiente de armazenamento das minúcias auxiliando a etapa de comparação. A estrutura do grafo capturada contém dados referentes à distribuição, tipo da minúcia e relacionamento entre as minúcias. O grafo formado foi representado em uma lista de adjacência onde o centróide foi ligado às n minúcias mais próximas, sendo este n configurável pelo usuário, conforme ilustrado na Figura 4.1.

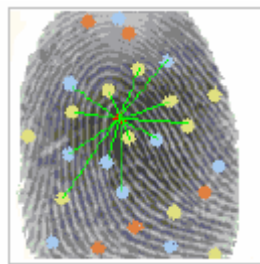


Figura 4.1: As minúcias representadas por círculos juntamente com o centróide sobrepostos à imagem da impressão digital (TEDESCO *et al*, 2001).

Na comparação dos grafos foram consideradas a posição (x,y) da minúcia armazenada no grafo, distância ao centro do grafo gerado e o tipo de minúcia (crista final ou bifurcação). Dessa forma, duas imagens são coincidentes quando localizadas na mesma posição em relação ao centro e são do mesmo tipo de padrão, aceitando no máximo 5% de variação. Os autores comentam que ao sofrer rotação alguns pontos da imagem não foram mapeados corretamente, ou até mesmo foram perdidos na etapa de pré-processamento e duas imagens distintas da mesma impressão digital originaram grafos distintos (Figura 4.2).

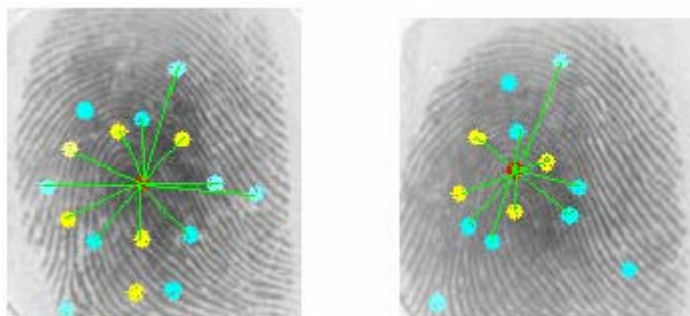


Figura 4.2: Dois grafos distintos para mesma impressão digital (TEDESCO *et al*, 2001).

Os resultados mais relevantes obtidos foram de uma taxa superior a 80% na comparação das impressões digitais com variação mínima de ângulos e de uma taxa superior a 50% onde as impressões digitais apresentavam variação de 45°.

Karu e Jain (1996) apresentaram um algoritmo de classificação da impressão digital utilizando a classificação exclusiva com cinco classes: *arch*, *tented arch*, *left loop*, *right loop* e *whorl*. O algoritmo extrai os pontos singulares (núcleos e deltas), ou seja, os aspectos globais da imagem da impressão digital e executa a classificação baseado no número de localizações dos pontos singulares detectados. Os autores afirmam que esse classificador é invariante à rotação, translação e pequenas mudanças de escala. Mas, não mencionam nenhum teste em imagens que sofreram rotação, translação ou mudanças de escala.

Maio e Maltoni (1996) desenvolveram uma abordagem estrutural de classificação da impressão digital cujo esquema funcional é mostrado na Figura 4.3.

Essa abordagem foi decomposta em cinco principais passos: computação da imagem direcional, segmentação da imagem direcional, construção de grafos relacionais, comparação e classificação dos grafos inexatos. Um algoritmo de agrupamento dinâmico é adotado para segmentar a imagem direcional com o objetivo de criar regiões tão homogêneas quanto possível. O algoritmo trabalha para minimizar a variação das direções dos elementos dentro das regiões e simultaneamente para manter a regularidade das formas da região. Começando pela segmentação da imagem direcional um grafo relacional é construído, criando para cada região um nó e para cada par de região adjacente um arco. Os autores afirmam que rotulando os nós e arcos do grafo, conseguem obter uma estrutura que resume as características topológicas da impressão digital e que é invariante a translação e rotação. Mas, não mencionam nenhum teste em imagens que sofreram rotação ou translação.

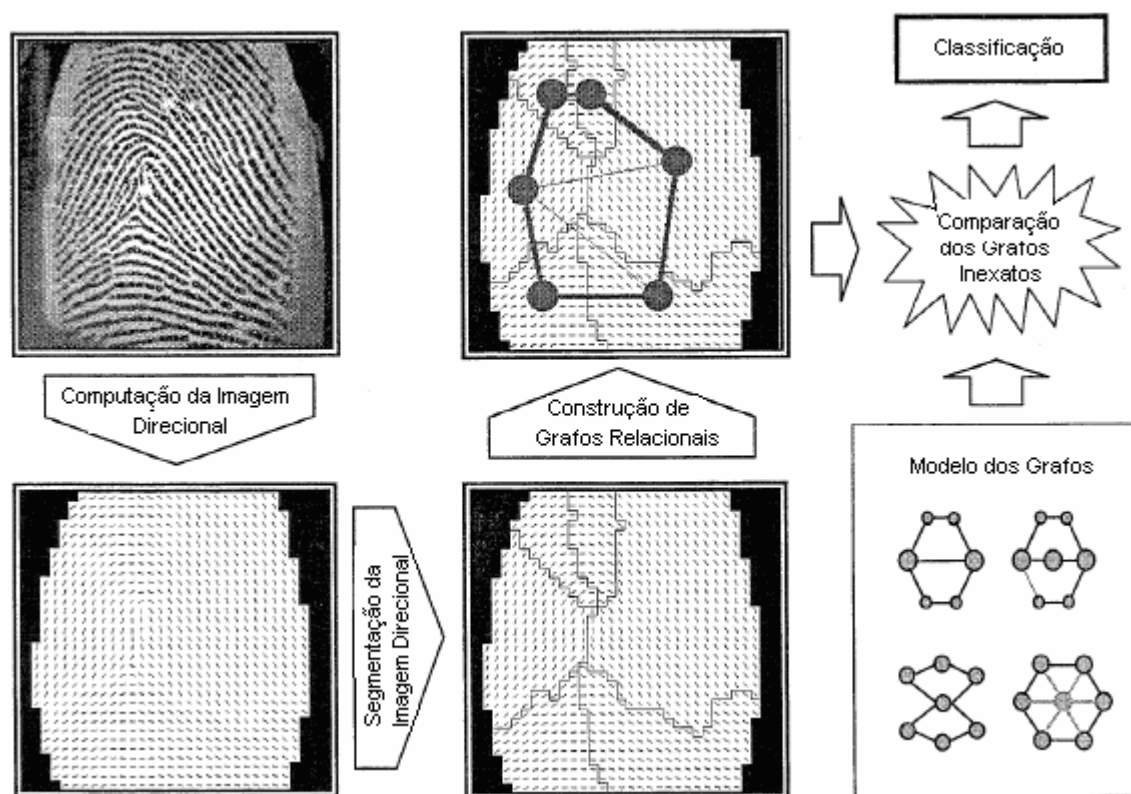


Figura 4.3: Os principais passos da abordagem. A figura mostra os resultados intermediários produzidos durante a classificação de uma impressão digital *Tented Arch* (MAIO, MALTONI, 1996).

Hrechak e Mchugh (1990) descreveram um modelo estrutural de impressões digitais, baseado nas relações estruturais locais entre características e um sistema de reconhecimento automático. Esse modelo foi projetado para permitir o reconhecimento da impressão mesmo na presença de distorções e pequenas rotações e translações. No processo de extração das características, as imagens foram selecionadas aleatoriamente e rotacionadas em 90, 180 e 270°. Estes testes resultaram características idênticas à imagem não rotacionada. Foi citado também, para outros ângulos de rotação, foi mais difícil simular efetivamente, porque a resolução das representações digitalizadas era limitada. Na comparação das impressões digitais e nos resultados experimentais apresentados, nenhum teste com as imagens rotacionadas foi citado.

Jain e outros (1997) introduziram um sistema de verificação de impressão digital *on-line* cujo propósito foi capturar imagens e compará-las com aquelas armazenadas em uma base de dados em tempo real. O sistema é composto de duas fases (Figura 4.4):

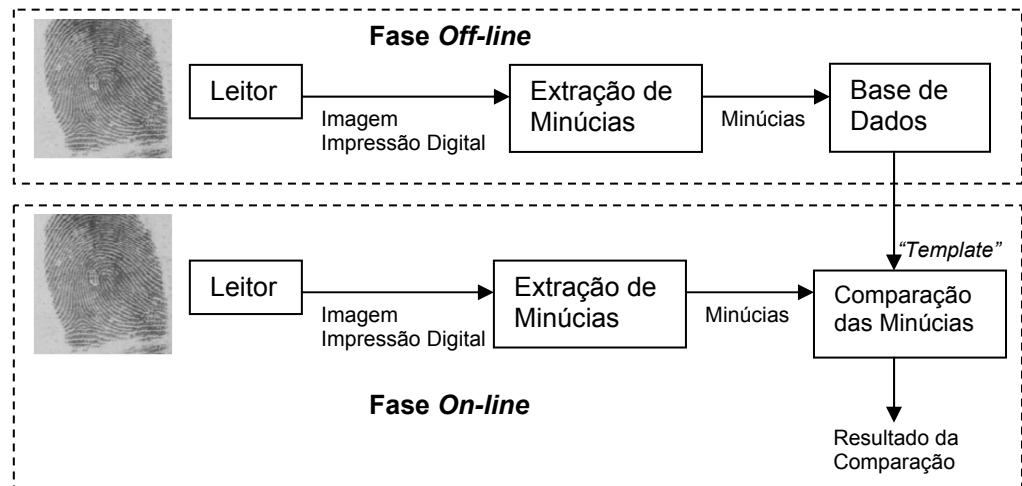


Figura 4.4: Sistema de Verificação de Impressão Digital *on-line* (JAIN et al, 1997)

- 1) *Fase off-line*: Várias imagens de impressões digitais de uma pessoa são capturadas e processadas por um módulo de extração de características. As características extraídas são armazenadas como *templates* na base de dados;
- 2) *Fase on-line*: O indivíduo informa sua identidade e posiciona o dedo no leitor de impressão digital. As minúcias são extraídas da imagem da impressão digital capturada; essas minúcias são fornecidas ao módulo de comparação, que compara-a novamente com seu próprio *template* na base de dados.

Os módulos de extração e comparação de minúcias são os componentes principais do sistema de verificação da impressão digital *on-line*. Uma versão modificada do algoritmo de extração de minúcias proposto por Ratha e outros (1995)

é usado no sistema, que é muito mais rápido e mais confiável. Um novo algoritmo de estimação de campo de orientação hierárquico resulta em um campo de orientação mais suave que melhora muito a performance da extração de minúcias. Um algoritmo baseado em alinhamento elástico é proposto para comparação das minúcias. Esse algoritmo é bem rápido, por que é capaz de encontrar as correspondências entre os pontos de minúcias sem utilizar uma busca exaustiva. Ao mesmo tempo, esse algoritmo tem uma boa performance, porque tem a habilidade de compensar adaptavelmente as deformações não-lineares e a posição inexata das transformações entre diferentes impressões digitais. Os resultados experimentais mostram que o sistema alcança uma performance excelente na operação ambiental realista, satisfazendo o tempo de resposta necessária para verificação *on-line*.

Com base nos resultados experimentais uma alta taxa de rejeitados ocorre devido a imagem de impressão digital de baixa qualidade como as mostradas na Figura 4.5. Os autores afirmam que podem melhorar esses resultados de comparação garantindo que a base de dados não contenha imagens de impressão digital com baixa qualidade. Nos resultados experimentais não mencionaram nenhum teste em imagens que sofreram rotação, translação ou mudança de escala, apenas concluíram que podem reduzir a taxa de rejeição pré-processando a base de dados para remover os *templates* de baixa qualidade armazenados.



Figura 4.5: Imagens de Impressão Digital de baixa qualidade (JAIN *et al*, 1997)

4.3. Considerações Finais

Diversos algoritmos para classificação de imagens de impressão digital foram propostos na literatura.

Neste capítulo, deu-se ênfase àqueles que procuraram resolver o problema relativo à rotação das imagens. O que se pode concluir é que em nenhuma situação o problema foi efetivamente resolvido. Ou seja, quando a imagem é rotacionada o índice de falsos positivos/negativos aumentam, reduzindo a eficiência dos algoritmos propostos.

O que se pretende com nosso trabalho é enfocar apenas o problema da rotação e propor um algoritmo que seja mais imune à esta variação considerando a mesma imagem de impressão digital em vários ângulos.

Capítulo 5

Metodologia

5.1. Metodologia

A Figura 5.1 mostra um diagrama de blocos com as principais etapas do algoritmo proposto.

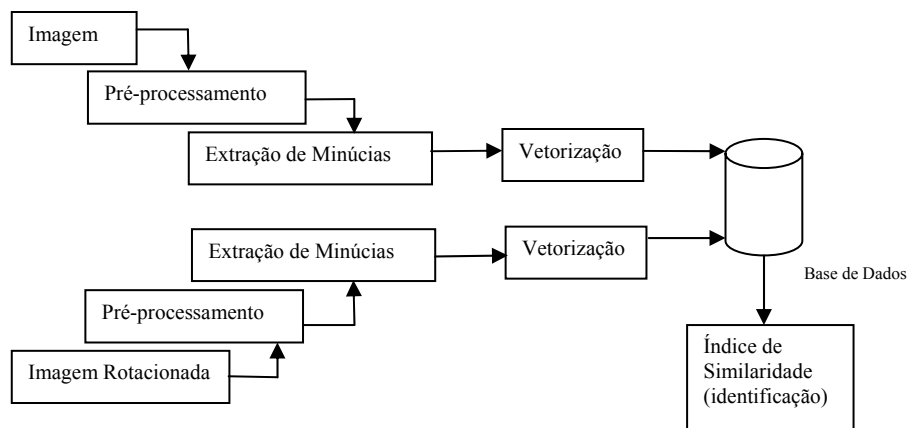


Figura 5.1: Diagrama da Metodologia Proposta

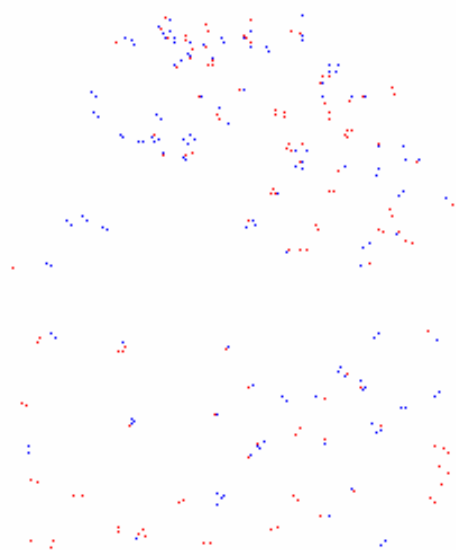
5.1.1. Pré-Processamento da Imagem

Para as estruturas das minúcias serem bem definidas, é necessário aplicar na imagem um detector de bordas. Foi utilizado o Detector de “*Canny*”, conforme visto no Capítulo 3, item 3.2, devido ao mesmo apresentar bons resultados com imagens de impressões digitais. A Figura 5.2 mostra o resultado obtido para uma imagem capturada.

**Imagem Original****Imagem Pré-Processada****Figura 5.2: Resultado do Pré-processamento da Imagem**

5.1.2. Extração de Minúcias

Após o pré-processamento, as minúcias são extraídas utilizando o método do Número de Cruzamentos (*Crossing Number* - CN) proposto por Raymond Thai, como visto no Capítulo 3, item 3.4 deste trabalho. A Figura 5.3 mostra o resultado da extração de minúcias em uma imagem exemplo.

**Minúcias Finais (vermelha) e Bifurcadas (azul)****Imagem pré-processada com Minúcias Finais e Bifurcadas****Figura 5.3: Extração de Minúcias Finais e Bifurcadas**

A extração de minúcias é uma abordagem de classificação contínua. Essa abordagem caracteriza cada impressão digital com um vetor numérico, cujos componentes denotem o grau da similaridade em relação a um conjunto pré-definido de protótipos de classe. Uma abordagem contínua permite impressões digitais ambíguas serem resolvidas e os esforços durante a recuperação da impressão digital serem equilibrados (CAPPELLI *et al*, 1999).

A extração de minúcias das imagens permite sintetizar propriedades inerentes da imagem, que serão utilizadas no processo de armazenagem e recuperação baseada em seu conteúdo.

Neste trabalho os vetores gerados pela extração das minúcias são separados por tipo (Figura 5.4) para minimizar o problema causado pela rotação das impressões digitais. Portanto, é gerado um vetor de minúcias do tipo final e outro vetor de minúcias do tipo bifurcada, e cada vetor contendo dados referentes ao quadrante, à coordenada x , coordenada y , e o tipo de minúcia.

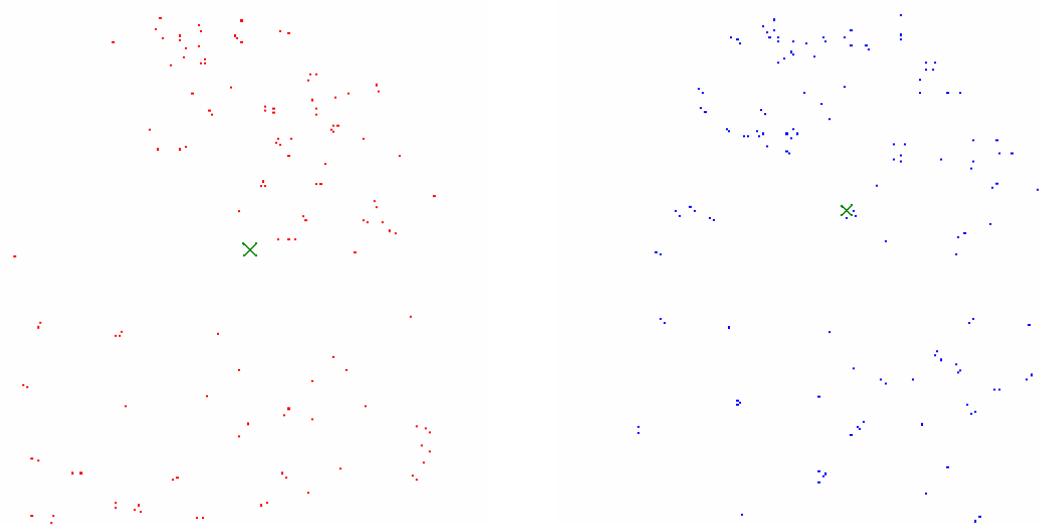


Figura 5.4: Minúcias Finais (vermelho), Bifurcadas (azul) e Centro de Massa (verde)

5.1.3. Proposta do Algoritmo de extração de características Invariantes com relação à Rotação.

Dada uma nuvem de pontos no espaço Cartesiano (x,y) pode-se calcular o Centro de Massa desta distribuição através da equação:

$$r_{cm} = \frac{m_1 r_1 + m_2 r_2 + m_3 r_3}{m_1 + m_2 + m_3} = \frac{\sum m_i r_i}{M} \quad (5.1)$$

Adotando-se a origem do Sistema de Coordenadas sobre este Centro de Massa conforme mostra a Figura 5.5, calcula-se a Distância Euclidiana de cada ponto da distribuição à sua origem obtendo-se o Vetor de Distâncias D1 obedecendo à ordem de varredura do primeiro ao quarto quadrante, da seguinte maneira:

No 1º Quadrante: x variando da origem até x máximo e y variando de y máximo até a origem.

No 2º Quadrante: y variando da origem até y mínimo e x variando de x máximo até a origem.

No 3º Quadrante: x variando da origem até x mínimo e y variando de y mínimo até a origem.

No 4º Quadrante: y variando da origem até y máximo e x variando de x mínimo até a origem.

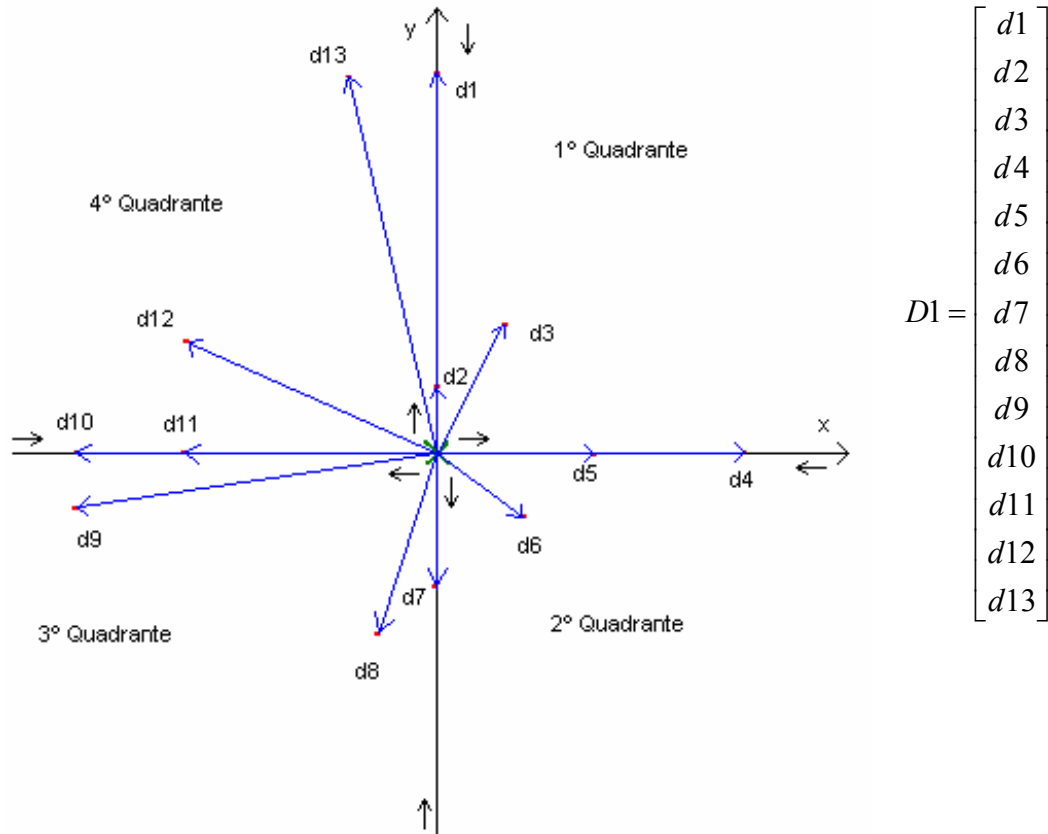


Figura 5.5: Geração do Vetor Distância ordenado do primeiro ao quarto quadrante

as outras impressões não rotacionadas verificando-se o menor valor da Distância Euclidiana entre cada Vetor.

5.1.4. Geração do Vetor de Características das minúcias

Para a implementação da metodologia proposta extrai-se as características das impressões digitais gerando-se um vetor de coordenadas das minúcias. As minúcias finais são separadas das bifurcadas, criando-se um vetor específico para cada tipo. O Centro de Massa da distribuição espacial de cada tipo de minúcia é calculado e armazenado no respectivo vetor. Considerando-se o Centro de Massa como origem do sistema de coordenadas, as Distâncias Euclidianas entre cada ponto de coordenada de minúcia e o respectivo Centro de Massa são calculadas e armazenadas no Vetor Distância, seguindo a ordem de quadrantes, como mostrado na Figura 5.7 e de acordo com o algoritmo proposto.

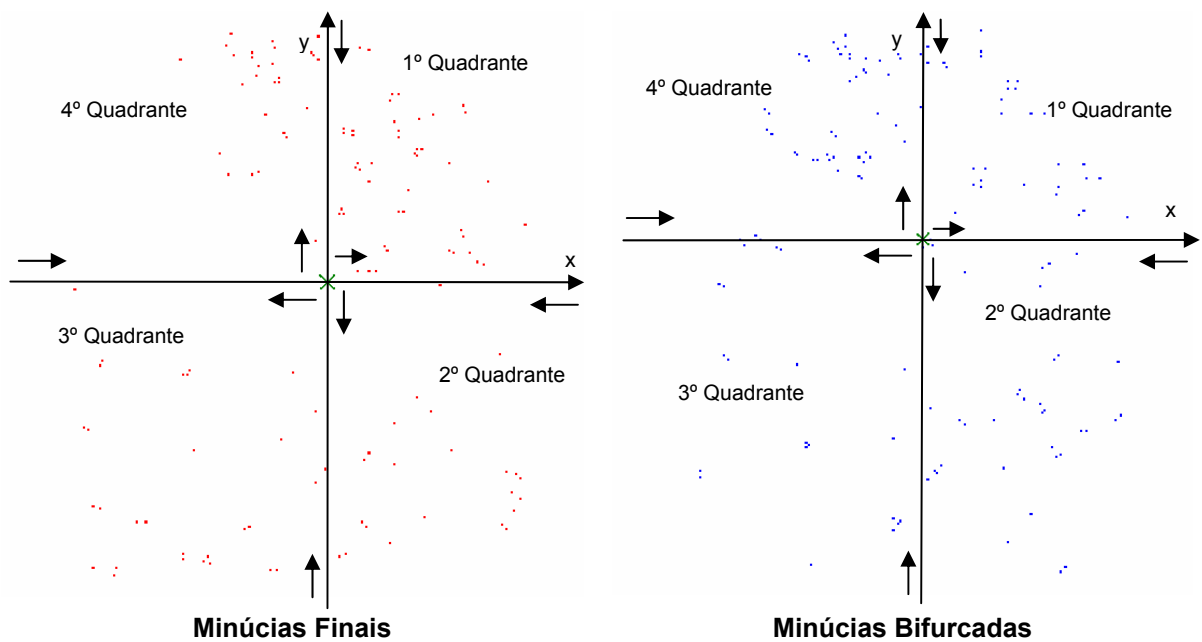


Figura 5.7: Minúcias com a origem do sistema de coordenadas deslocada para o Centro de Massa

O Vetor Distância por tipo de minúcia conterà, através de uma varredura nas coordenadas no sentido horário do primeiro ou quarto quadrante, as minúcias estáveis, que estão localizadas com valor da Distância Euclidiana ao Centro de

Massa superior a 10 e inferior a 80. Esta restrição foi adotada devido ao fato observado que as minúcias próximas ao Centro de Massa sofrem distorções de posicionamento quando a rotação da imagem é realizada, pois os softwares utilizam o centro da imagem para rotacioná-la, alterando a localização real das minúcias. As minúcias localizadas com distância superior a 80 estão muito próximas da borda da imagem, onde ocorrem problemas de falhas, interrupção das cristas ou distorções durante a aquisição da imagem, sendo identificadas como falsas minúcias (Figura 5.8).

Esta forma de organização do Vetor Distância permite gerar o mesmo vetor rotacionado para a mesma impressão digital, rotacionada de um determinado ângulo.

Através de sucessivas rotações do Vetor Distância estabelece-se a menor distância entre os vetores da Base de Imagens. Teoricamente, se nenhum ruído ou distorção influenciar o processamento, é possível ter distância zero entre dois vetores de uma mesma imagem rotacionada, utilizando a Distância de Euclidiana que retornará zero se ambos os vetores forem idênticos ou um valor positivo que aumentará quanto maior a distância (ou dissimilaridade) entre os vetores, ou melhor, entre as Impressões Digitais comparadas.

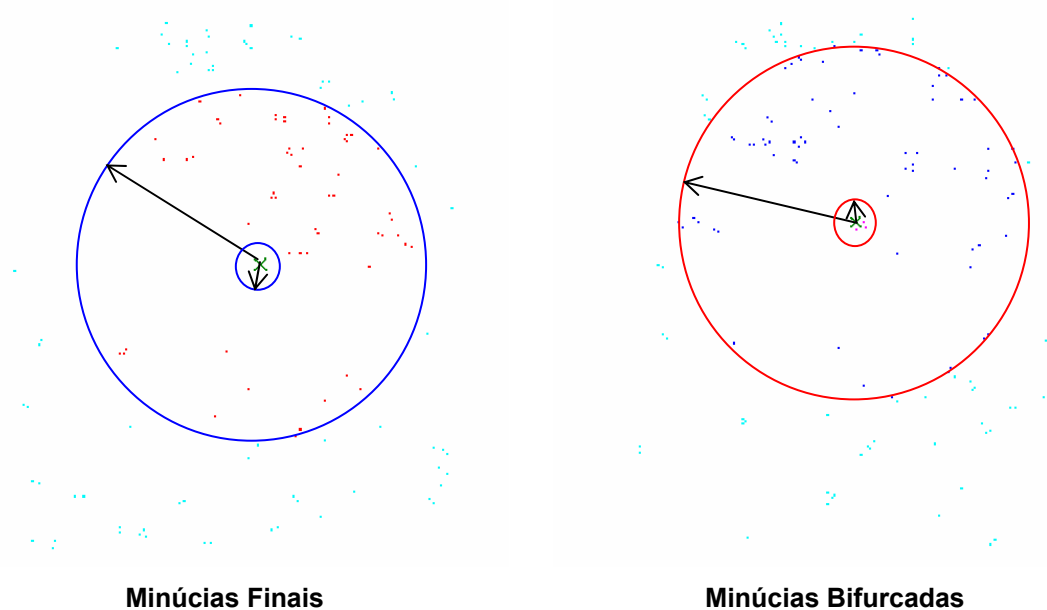


Figura 5.8: Minúcias com valor da Distância Euclidiana ao Centro de Massa superior a 10 e inferior a 80

5.1.5. Determinação do Vetor de Alinhamento

Para realizar sucessivas rotações do Vetor Distância da impressão digital rotacionada, determina-se um Vetor de Alinhamento com o objetivo de ordenar os pontos de distâncias de minúcias do tipo final de linha e do tipo bifurcada. Esta ordenação é importante para o cálculo da Distância Euclidiana que determinará a similaridade ou dissimilaridade entre os vetores, ou melhor entre as Impressões Digitais comparadas.

O Vetor de Alinhamento conterà quatro posições do Vetor Distância da imagem original que serão comparadas sucessivas vezes com o Vetor Distância da imagem rotacionada de quatro em quatro posições. Para esta comparação calcula-se a diferença entre as suas médias, buscando-se o menor valor (maior alinhamento). Este será o ponto de partida (índice do vetor) de rotação para a re-ordenação do Vetor Distância da impressão digital rotacionada.

A utilização de um Vetor de Alinhamento de dimensão inferior a quatro, resultou em pontos de partida falsos, dificultando a ordenação. As distâncias calculadas entre o Vetor Distância da imagem original e o Vetor Distância da imagem rotacionada do tipo final de linha mostraram valores próximos de zero. As distâncias calculadas entre o Vetor Distância da imagem original e o Vetor Distância da imagem rotacionada do tipo bifurcada apresentaram valores muito elevados.

As minúcias espúrias prejudicam a localização do ponto de partida da ordenação, por definir pontos de partida falsos. Para melhorar a localização foram definidas seis combinações de quatro posições do Vetor Distância da imagem original para depois ser efetuada a comparação com o Vetor Distância da imagem rotacionada, da seguinte maneira:

- 1ª) posição1, posição2, posição3, posição4
- 2ª) posição1, posição2, posição4, posição5
- 3ª) posição1, posição3, posição4, posição5
- 4ª) posição2, posição3, posição4, posição5
- 5ª) posição1, posição2, posição3, posição5
- 6ª) posição3, posição4, posição5, posição6

Para cada combinação é gerado um novo vetor com a diferença entre as médias dos valores dos Vetores de Alinhamento, ordenando-se esse novo vetor em ordem crescente. Os pontos de partida que não ultrapassarem o valor de limiar estabelecido de 2.5 são validados por outra função que compara o Vetor Distância de minúcias da imagem original com o Vetor Distância de minúcias da imagem rotacionada, iniciando-se de cada ponto de partida encontrado. A melhor combinação será a que na validação conseguir localizar quatro minúcias respeitando-se a combinação. Se nenhuma combinação for validada, refaz-se o Vetor de Alinhamento com outras seis combinações de quatro posições partindo-se da 5ª posição em diante repetindo-se todo o processo de validação, ou seja:

- 7ª) posição5, posição6, posição7, posição8
- 8ª) posição5, posição6, posição8, posição9
- 9ª) posição5, posição7, posição8, posição9
- 10ª) posição6, posição7, posição8, posição9
- 11ª) posição5, posição6, posição7, posição9
- 12ª) posição7, posição8, posição9, posição10

O ponto de partida não será localizado caso as imagens não sejam similares ou quando coincidir ter uma ou várias minúcias espúrias em todos os casos de combinação.

Como exemplo considere o Vetor Distância A contendo as distâncias das minúcias ao Centro de Massa da imagem original e o Vetor Distância B contendo as distâncias das minúcias ao Centro de Massa da imagem rotacionada, conforme mostra a Figura 5.9.

A primeira combinação para o Vetor de Alinhamento é baseada no Vetor Distância A e definida pela posição1, posição2, posição3, posição4, representada pelos valores, 56.223, 41.785, 70.88 e 71.021 respectivamente (ver Figura 5.9). Calcula-se a média deste Vetor de Alinhamento, que resulta em 59.9772. Compara-se, então, com o Vetor Distância B sucessivas vezes de quatro em quatro posições, armazenando-se a diferença entre as médias em um novo vetor, ordenando-se o mesmo em ordem crescente (Figura 5.10).

$A =$	56.223	posição1	$B =$	71.84	$(71.84 + 67.543 + 76.158 + 78.645)/4 = 73.5465$
	41.785	posição2		67.543	$(67.543 + 76.158 + 78.645 + 56.223)/4 = 69.6422$
	70.88	posição3		76.158	$(76.158 + 78.645 + 56.223 + 70.88)/4 = 70.4765$
	71.021	posição4		78.645	$(78.645 + 56.223 + 70.88 + 71.021)/4 = 69.1922$
	72.945	posição5		56.223	$(56.223 + 70.88 + 71.021 + 72.945)/4 = 67.7672$
	74.626	posição6		70.88	$(70.88 + 71.021 + 72.945 + 74.626)/4 = 72.368$
	18.682	posição7		71.021	$(71.021 + 72.945 + 74.626 + 18.682)/4 = 59.3185$
	63.702	posição8		72.945	$(72.945 + 74.626 + 18.682 + 63.702)/4 = 57.4887$
	57.428	posição9		74.626	$(74.626 + 18.682 + 63.702 + 57.428)/4 = 53.6095$
	56.604	posição10		18.682	$(18.682 + 63.702 + 57.428 + 56.604)/4 = 49.104$
	73.11	posição11		63.702	$(63.702 + 57.428 + 56.604 + 71.84)/4 = 62.3935$
	65.521	posição12		57.428	$(57.428 + 56.604 + 71.84 + 67.543)/4 = 63.3537$
	67.72	posição13		56.604	$(56.604 + 71.84 + 67.543 + 76.158)/4 = 68.0362$

Figura 5.9: Vetor Distância A (original) e Vetor Distância B (rotacionada de um ângulo θ)

$B =$	71.84	$(71.84 + 67.543 + 76.158 + 78.645)/4 = 73.5465$	$novoVetor =$	1	13.5692	$novoVetor =$	7	0.6587
	67.543	$(67.543 + 76.158 + 78.645 + 56.223)/4 = 69.6422$		2	9.665		11	2.4163
	76.158	$(76.158 + 78.645 + 56.223 + 70.88)/4 = 70.4765$		3	10.4993		8	2.4885
	78.645	$(78.645 + 56.223 + 70.88 + 71.021)/4 = 69.1922$		4	9.215		12	3.3765
	56.223	$(56.223 + 70.88 + 71.021 + 72.945)/4 = 67.7672$		5	7.79		9	6.3677
	70.88	$(70.88 + 71.021 + 72.945 + 74.626)/4 = 72.368$		6	12.3908		5	7.79
	71.021	$(71.021 + 72.945 + 74.626 + 18.682)/4 = 59.3185$		7	0.6587		13	8.059
	72.945	$(72.945 + 74.626 + 18.682 + 63.702)/4 = 57.4887$		8	2.4885		4	9.215
	74.626	$(74.626 + 18.682 + 63.702 + 57.428)/4 = 53.6095$		9	6.3677		2	9.665
	18.682	$(18.682 + 63.702 + 57.428 + 56.604)/4 = 49.104$		10	10.8732		3	10.4993
	63.702	$(63.702 + 57.428 + 56.604 + 71.84)/4 = 62.3935$		11	2.4163		10	10.8732
	57.428	$(57.428 + 56.604 + 71.84 + 67.543)/4 = 63.3537$		12	3.3765		6	12.3908
	56.604	$(56.604 + 71.84 + 67.543 + 76.158)/4 = 68.0362$		13	8.059		1	13.5692

Figura 5.10: Comparação do Vetor de Alinhamento da primeira combinação do Vetor Distância A com o Vetor Distância B, gerando novo vetor e novo vetor ordenado

Após a ordenação consideram-se os pontos de partida que não ultrapassem o valor especificado de limiar de 2.5, representados na Figura 5.10 pelo primeiro, segundo e terceiro elementos do novo vetor ordenado. Validam-se os três elementos, respeitando-se o ponto de partida, através de outra função que compara o Vetor Distância A com o Vetor Distância B. A melhor combinação será a que na validação conseguir localizar as quatro minúcias respeitando-se a combinação (Figura 5.11).

$B =$	71.84	<i>posição1</i>		56.223
	67.543	<i>posição2</i>		41.785
	76.158	<i>posição3</i>		70.88
	78.645	<i>posição4</i>		71.021
	56.223	<i>posição5</i>		72.945
	70.88	<i>posição6</i>		74.626
	71.021	<i>posição7</i> ←	$A =$	18.682
	72.945	<i>posição8</i> ←		63.702
	74.626	<i>posição9</i>		57.428
	18.682	<i>posição10</i>		56.604
	63.702	<i>posição11</i> ←		73.11
	57.428	<i>posição12</i>		65.521
	56.604	<i>posição13</i>		67.72

← Validando-se a primeira combinação do Vetor Distância A iniciando-se pelo ponto de partida 7 no Vetor Distância B, nenhuma distância de minúcia foi localizada, respeitando-se o limiar de 2.5.

← Validando-se a primeira combinação do Vetor Distância A iniciando-se pelo ponto de partida 11 no Vetor Distância B, localiza-se apenas a distância de minúcia 71.021 do Vetor Distância A na posição1 do Vetor Distância B, respeitando-se o limiar de 2.5.

← Validando-se a primeira combinação do Vetor Distância A iniciando-se pelo ponto de partida 8 no Vetor Distância B, nenhuma distância de minúcia foi localizada respeitando-se o limiar de 2.5.

Figura 5.11: Validação dos três elementos do novo vetor que não ultrapassaram o valor de limiar 2.5 da primeira combinação do Vetor Distância A

A segunda combinação do Vetor Distância A é definida pela posição1, posição2, posição4, posição5, representada pelos valores, 56.223, 41.785, 71.021 e 72.945 respectivamente (ver Figura 5.9). Calcula-se a média deste Vetor de Alinhamento, que resulta em 60.4935. Repete-se então o mesmo procedimento (Figura 5.12).

$B =$	71.84	$(71.84 + 67.543 + 76.158 + 78.645) / 4 = 73.5465$	$novoVetor =$	1	13.053	$novoVetor =$	7	1.175
	67.543	$(67.543 + 76.158 + 78.645 + 56.223) / 4 = 69.6422$		2	9.1487		11	1.9
	76.158	$(76.158 + 78.645 + 56.223 + 70.88) / 4 = 70.4765$		3	9.983		12	2.8602
	78.645	$(78.645 + 56.223 + 70.88 + 71.021) / 4 = 69.1922$		4	8.6987		8	3.0048
	56.223	$(56.223 + 70.88 + 71.021 + 72.945) / 4 = 67.7672$		5	7.2437		9	6.884
	70.88	$(70.88 + 71.021 + 72.945 + 74.626) / 4 = 72.368$		6	11.8745		5	7.2437
	71.021	$(71.021 + 72.945 + 74.626 + 18.682) / 4 = 59.3185$		7	1.175		13	7.8685
	72.945	$(72.945 + 74.626 + 18.682 + 63.702) / 4 = 57.4887$		8	3.0048		4	8.6987
	74.626	$(74.626 + 18.682 + 63.702 + 57.428) / 4 = 53.6095$		9	6.884		2	9.1487
	18.682	$(18.682 + 63.702 + 57.428 + 56.604) / 4 = 49.104$		10	11.3895		3	9.983
	63.702	$(63.702 + 57.428 + 56.604 + 71.84) / 4 = 62.3935$		11	1.9		10	11.3895
	57.428	$(57.428 + 56.604 + 71.84 + 67.543) / 4 = 63.3537$		12	2.8602		6	11.8745
	56.604	$(56.604 + 71.84 + 67.543 + 76.158) / 4 = 68.0362$		13	7.8685		1	13.053

Figura 5.12: Comparação do Vetor de Alinhamento da segunda combinação do Vetor Distância A com o Vetor Distância B, gerando novo vetor e novo vetor ordenado

Após a ordenação consideram-se os pontos de partida que não ultrapassarem o valor especificado de limiar de 2.5, representados na Figura 5.12 pelo primeiro e segundo elementos do novo vetor ordenado. Validam-se os dois elementos

respeitando-se o ponto de partida, através de outra função que compara o Vetor Distância A com o Vetor Distância B. A melhor combinação será a que na validação conseguir localizar as quatro minúcias respeitando-se a combinação utilizada (Figura 5.13).

$B =$	71.84	<i>posição1</i>	56.223	←	← Validando-se a segunda combinação do Vetor Distância A iniciando-se pelo ponto de partida 7 no Vetor Distância B, localiza-se apenas a distância de minúcia 72.945 do Vetor Distância A na posição1 do Vetor Distância B respeitando-se o limiar de 2.5.
	67.543	<i>posição2</i>	41.785	←	
	76.158	<i>posição3</i>	70.88		
	78.645	<i>posição4</i>	71.021	←	
	56.223	<i>posição5</i>	72.945	←	
	70.88	<i>posição6</i>	74.626		
	71.021	<i>posição7</i> ←	18.682		← Validando-se a segunda combinação do Vetor Distância A iniciando-se pelo ponto de partida 11 no Vetor Distância B, localiza-se apenas a distância de minúcia 72.945 do Vetor Distância A na posição1 do Vetor Distância B respeitando-se o limiar de 2.5.
	72.945	<i>posição8</i>	63.702		
	74.626	<i>posição9</i>	57.428		
	18.682	<i>posição10</i>	56.604		
	63.702	<i>posição11</i> ←	73.11		
	57.428	<i>posição12</i>	65.521		
	56.604	<i>posição13</i>	67.72		
			$A =$		

Figura 5.13: Validação dos três elementos do novo vetor que não ultrapassaram o valor de limiar 2.5 da segunda combinação do Vetor Distância A

Na validação da segunda combinação, iniciando-se pelo ponto de partida 11 no Vetor Distância B, foi localizada uma distância de minúcias. No entanto, a melhor validação considera a localização de pelos menos quatro distâncias de minúcias. Como o resultado não foi atingido, realiza-se a terceira combinação de posições.

A terceira combinação do Vetor Distância A é definida pela posição1, posição3, posição4, posição5, representada pelos valores, 56.223, 70.88, 71.021 e 72.945 respectivamente (ver Figura 5.9). Calcula-se a média deste Vetor de Alinhamento, que resulta em 67.7672. Repete-se o procedimento (Figura 5.14).

$B =$	71.84	$(71.84 + 67.543 + 76.158 + 78.645)/4 = 73.5465$	$novoVetor =$	1	5.7793	$novoVetor =$	5	0
	67.543	$(67.543 + 76.158 + 78.645 + 56.223)/4 = 69.6422$		2	1.875		13	0.269
	76.158	$(76.158 + 78.645 + 56.223 + 70.88)/4 = 70.4765$		3	2.7093		4	1.425
	78.645	$(78.645 + 56.223 + 70.88 + 71.021)/4 = 69.1922$		4	1.425		2	1.875
	56.223	$(56.223 + 70.88 + 71.021 + 72.945)/4 = 67.7672$		5	0		3	2.7093
	70.88	$(70.88 + 71.021 + 72.945 + 74.626)/4 = 72.368$		6	4.6008		12	4.4135
	71.021	$(71.021 + 72.945 + 74.626 + 18.682)/4 = 59.3185$		7	8.4487		6	4.6008
	72.945	$(72.945 + 74.626 + 18.682 + 63.702)/4 = 57.4887$		8	10.2785		11	5.3737
	74.626	$(74.626 + 18.682 + 63.702 + 57.428)/4 = 53.6095$		9	14.1577		1	5.7793
	18.682	$(18.682 + 63.702 + 57.428 + 56.604)/4 = 49.104$		10	18.6632		7	8.4487
	63.702	$(63.702 + 57.428 + 56.604 + 71.84)/4 = 62.3935$		11	5.3737		8	10.2785
	57.428	$(57.428 + 56.604 + 71.84 + 67.543)/4 = 63.3537$		12	4.4135		9	14.1577
	56.604	$(56.604 + 71.84 + 67.543 + 76.158)/4 = 68.0362$		13	0.269		10	18.6632

Figura 5.14: Comparação do Vetor de Alinhamento da terceira combinação do Vetor Distância A com o Vetor Distância B, gerando novo vetor e novo vetor ordenado

Após a ordenação consideram-se os pontos de partida que não ultrapassarem o valor especificado de limiar de 2.5, representados na Figura 5.14 pelo primeiro, segundo, terceiro e o quarto elementos do novo vetor ordenado. Validam-se os quatro elementos respeitando o ponto de partida, por outra função que compara o Vetor Distância A com o Vetor Distância B. A melhor combinação será a que na validação conseguir localizar pelo menos as quatro distâncias de minúcias respeitando-se a combinação (Figura 5.15).

$B =$	71.84	<i>posição1</i>	56.223	←	← Validando-se a terceira combinação do Vetor Distância A iniciando-se pelo ponto de partida 5 no Vetor Distância B, respeitando-se o limiar de 2.5, localiza-se as distâncias de minúcias 56.223, 70.88, 71.021 e 72.945 do Vetor Distância A na posição5, posição6, posição7 e posição8, respectivamente, encerrando-se a validação.
	67.543	<i>posição2</i> ←	41.785		
	76.158	<i>posição3</i>	70.88	←	
	78.645	<i>posição4</i> ←	71.021	←	
	56.223	<i>posição5</i> ←	72.945	←	
	70.88	<i>posição6</i>	74.626		
	71.021	<i>posição7</i>	18.682		
	72.945	<i>posição8</i>	63.702		
	74.626	<i>posição9</i>	57.428		
	18.682	<i>posição10</i>	56.604		
	63.702	<i>posição11</i>	73.11		
	57.428	<i>posição12</i>	65.521		
	56.604	<i>posição13</i> ←	67.72		

Figura 5.15: Validação dos quatros elementos do novo vetor que não ultrapassaram o valor de limiar 2.5 da terceira combinação do Vetor Distância A

Na validação da terceira combinação iniciando-se pelo ponto de partida 5 no Vetor Distância B, foram localizadas as quatro distâncias de minúcias necessárias, encerrando-se o processo. O Vetor Distância B é então rotacionado a partir do ponto de partida 5, ou seja, de quatro posições alinhando-se os dois Vetores Distâncias (A e B).

5.1.6. Minúcias Espúrias

Após a localização do ponto de partida, ordena-se o Vetor Distância da imagem rotacionada, separando-se as minúcias espúrias tanto da imagem original como as da imagem rotacionada em outros vetores, chamados de Vetor Espúrio Origem e Vetor Espúrio Rotacionado, respectivamente. Observou-se que algumas das minúcias espúrias encontradas no vetor da imagem original também existiam no vetor da imagem rotacionada, pertencentes aos mesmos quadrantes e com valor da Distância Euclidiana do ponto de minúcia ao Centro de Massa próximos. Foi então desenvolvida uma nova função de re-inclusão dessas minúcias espúrias no Vetor Distância da imagem origem e no Vetor da imagem rotacionada, conforme mostra a Figura 5.16.

$$\begin{array}{l}
 \text{VetorEspúrio Registro} = \begin{bmatrix} 1 & 44.283 & 1 & 25 \\ 2 & 42.426 & 1 & 26 \\ 3 & 59.034 & 1 & 27 \\ 5 & 39.825 & 1 & 28 \\ 42 & 79.605 & 4 & 14 \end{bmatrix}
 \end{array}
 \qquad
 \begin{array}{l}
 \text{VetorEspúrio Autenticação} = \begin{bmatrix} 25 & 57.697 & 2 & 2 \\ 26 & 35.384 & 2 & 3 \\ 35 & 78.873 & 2 & 14 \\ 40 & 10.05 & 4 & 18 \\ 3 & 69.584 & 1 & 33 \\ 21 & 79.006 & 2 & 49 \\ 22 & 43.186 & 2 & 49 \\ 23 & 41.304 & 2 & 49 \\ 24 & 79.831 & 2 & 49 \end{bmatrix}
 \end{array}$$

Figura 5.16: Vetor Espúrio de Registro e Vetor Espúrio de Autenticação

O Vetor Espúrio de Registro armazena na primeira coluna a seqüência da minúcia no Vetor Distância de Registro, na segunda coluna a Distância Euclidiana da minúcia ao Centro de Massa, na terceira coluna o quadrante desta minúcia e na quarta coluna a seqüência com a qual esta minúcia deveria estar localizada no Vetor Distância de Autenticação.

O Vetor Espúrio de Autenticação armazena na primeira coluna a seqüência da minúcia no Vetor Distância de Autenticação, na segunda coluna a Distância Euclidiana da minúcia ao Centro de Massa, na terceira coluna o quadrante desta minúcia e na quarta coluna a seqüência que esta minúcia deveria estar localizada no Vetor Distância de Registro.

Observando-se estes Vetores Espúrios pode-se encontrar minúcias espúrias do Vetor Espúrio de Registro no Vetor Espúrio de Autenticação realizando sucessivas rotações, verificando se a diferença das Distâncias Euclidiana não ultrapassa o valor de limiar de 2.5 e a diferença entre as seqüências no Vetor Distância de Autenticação não ultrapasse o limiar 5. No exemplo da Figura 5.16, pode-se encontrar as minúcias espúrias do Vetor Espúrio de Registro definidas pelas Distâncias Euclidiana 44.283, 42.426 e 59.034 e respectivas seqüências do Vetor Distância de Autenticação 25, 26 e 27 no Vetor Espúrio de Autenticação realizando sucessivas rotações, localizando-se as Distâncias Euclidianas 43.186, 41.304 e 57.697 das respectivas seqüências do Vetor Distância de Autenticação 22, 23 e 25. Conclui-se que houve um pequeno deslocamento e que estas minúcias poderão ser re-incluídas nos Vetores Distância de Registro e de Autenticação.

O problema observado com os Vetores Distâncias das minúcias referentes à Impressão Digital rotacionada, é um problema de Reconhecimento de Padrões com Dados Faltantes ou Espúrios (Dixon, 1979).

Estes Vetores Espúrios são importantes para a identificação da impressão digital. Dixon (1979) descreve que faz sentido aplicar a intuição humana para aumentar a percepção nos problemas dos dados faltantes. Como exemplo considere os vetores A e B:

A = (1.2 3.7 10.9 6.3 5.9)

B = (1.1 3.5 espúria 6.2 5.7)

Pode-se concluir que aparentemente o dado faltante será próximo de 10.9 desde que todos os outros elementos estejam próximos. Mais precisamente, um outro meio de estimar o dado faltante é que será menor que 10.9, desde que todos os elementos de B sejam menores do que os elementos correspondentes em A. Existem vários caminhos para interpretar essa sensibilidade intuitiva:

- 1) Uma suposição que os dados são agrupados.

- 2) Uma suposição que as diversas características são correlacionadas, por isso pode-se utilizar uma interpolação linear sobre linhas ou colunas.
- 3) Uma suposição que se as diferenças são pequenas ao longo de quatro dimensões, elas também serão pequenas ao longo de cinco.

Com esta analogia Dixon (1979) descreveu diversas técnicas simples e de fácil implementação. Estas técnicas eliminam parte dos dados, substituem os dados faltantes por valores estimados, ou calculam distância entre dois vetores que contém dados faltantes. Algumas destas técnicas são:

- 1) Simplesmente apagar os dados faltantes ou substituir pelo valor zero (0). Esta técnica conduz para uma perda de informação e apenas deveria ser usada em casos onde a quantidade de dados faltantes seja muito pequena.
- 2) Substituir o dado faltante pela média entre os k vizinhos mais próximos. No exemplo dos vetores A e B, suponha-se que o 3º elemento de B é faltante, logo a média entre os elementos de A “3.7” ; ”10.9” ; “6.3” é “6.9”, substituindo o dado faltante pela média dos dois vizinhos mais próximos, o vetor B resultaria em:

$$A = (1.2 \quad \mathbf{3.7} \quad \mathbf{10.9} \quad \mathbf{6.3} \quad 5.9)$$

$$B = (1.1 \quad 3.5 \quad \mathbf{6.9} \quad 6.2 \quad 5.7)$$

- 3) Calcular a distância d_j entre os dois vetores x_j e x_k que contém dados faltantes utilizando a seguinte condição:

$$d_j = \begin{cases} 0 & \longrightarrow \text{se } x_j \text{ ou } x_k \text{ é dado faltante} \\ x_j - x_k & \longrightarrow \text{caso contrário} \end{cases} \quad (5.2)$$

- E a distância entre x_j e x_k é definida como:

$$d(j, k) = \frac{d}{d - d_o} \sum d_j^2 \quad (5.3)$$

Onde d_o é a quantidade de dados faltantes em x_j ou x_k ou nos dois e d é a quantidade de dados.

Dixon (1979) conclui que as técnicas 2 e 3 são excelentes técnicas e que a escolha depende das características dos dados e facilidade de implementação. E que a técnica 3 é um pouco mais consistente na performance do que a técnica 2.

Neste trabalho foram implementadas as técnicas 2 e 3 para tentar solucionar os problemas de minúcias espúrias e garantir um maior desempenho do algoritmo.

Capítulo 6

Discussão dos Resultados

6.1. Discussão dos Resultados

Resultados experimentais foram obtidos com imagens na posição original e a mesma imagem rotacionada em 90, 180 e 270°. Foram utilizadas 672 imagens de impressão digital em níveis de cinza no tamanho de 256 x 256 *pixels*, provenientes de 21 pessoas diferentes com 8 aquisições de cada uma, com diferentes problemas na captura, como por exemplo, pressão, posição, oleosidade do dedo, poeira no leitor, marcas de aquisição anteriores, etc (MALTONI *et al*, 2003).

Na extração das minúcias são gerados vetores que, para imagens em posição original, serão chamados de **Vetores Distância de Registro** e para imagens rotacionadas serão chamados de **Vetores Distância de Autenticação**.

A Figura 6.1 mostra as imagens de uma impressão digital na posição original e a mesma imagem rotacionada de 180°.



Imagem Original

Imagem Rotacionada de 180°

Figura 6.1: Imagem Original de Impressão Digital e a mesma Imagem Rotacionada de 180°

A Figura 6.2 mostra as mesmas imagens da figura 6.1 após o pré-processamento.



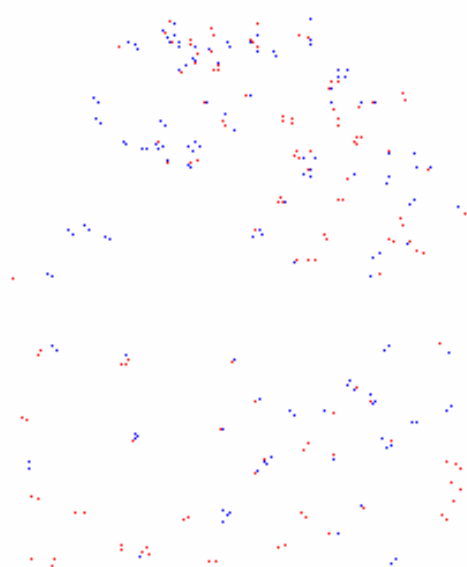
Imagem Original



Imagem Rotacionada de 180°

Figura 6.2: Imagens Pré-processadas de Impressão Digital

As Figuras 6.3 e 6.4 mostram as imagens da impressão digital original e rotacionada de 180°, respectivamente, com as minúcias finais (vermelho) e bifurcadas (azul).



Minúcias Finais (vermelha) e Bifurcadas (azul)

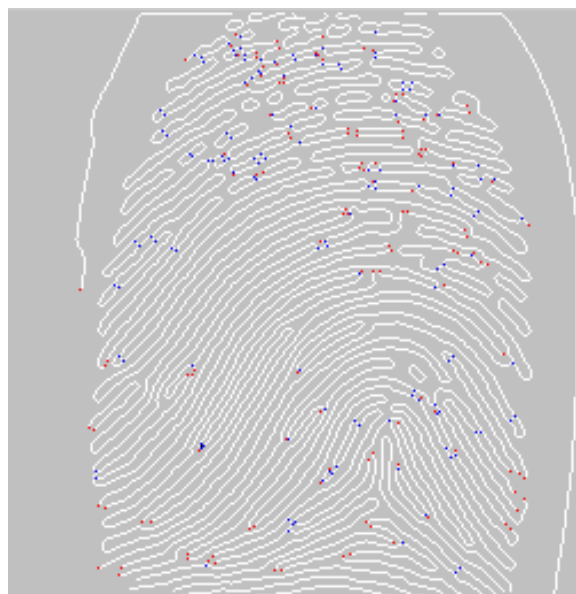


Imagem pré-processada com Minúcias Finais e Bifurcadas

Figura 6.3: Extração de Minúcias Finais e Bifurcadas da Imagem Original

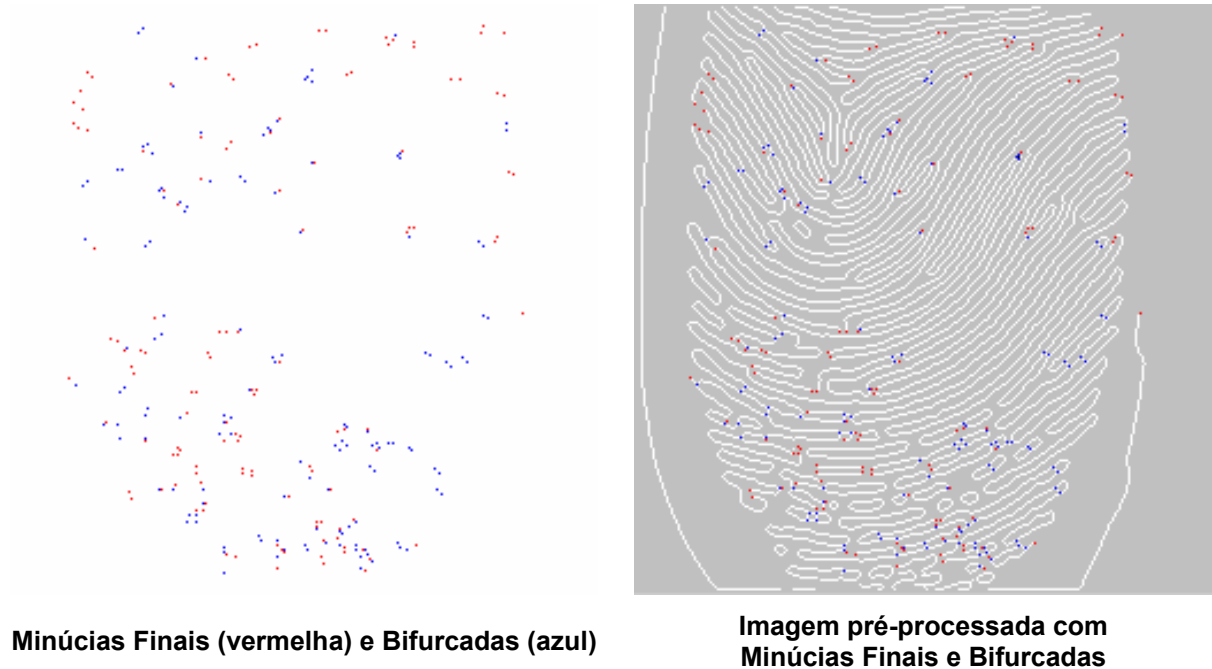


Figura 6.4: Extração de Minúcias Finais e Bifurcadas da Imagem Rotacionada de 180°

Após a extração as minúcias finais são separadas das minúcias bifurcadas (Figura 6.5 para Imagem Original e Figura 6.6 para Imagem Rotacionada de 180°), sendo que cada tipo de minúcia terá um vetor específico, e o Centro de Massa do conjunto de cada tipo de minúcia é calculado e armazenado no referente vetor. Considerando-se o Centro de Massa como origem do sistema de coordenadas, a Distância Euclidiana entre cada ponto de minúcia e o Centro de Massa é calculada e armazenada no Vetor Distância. As minúcias estáveis estão localizadas com valor de Distância Euclidiana ao Centro de Massa superior a 10 e inferior a 80, seguindo a ordem de quadrantes.

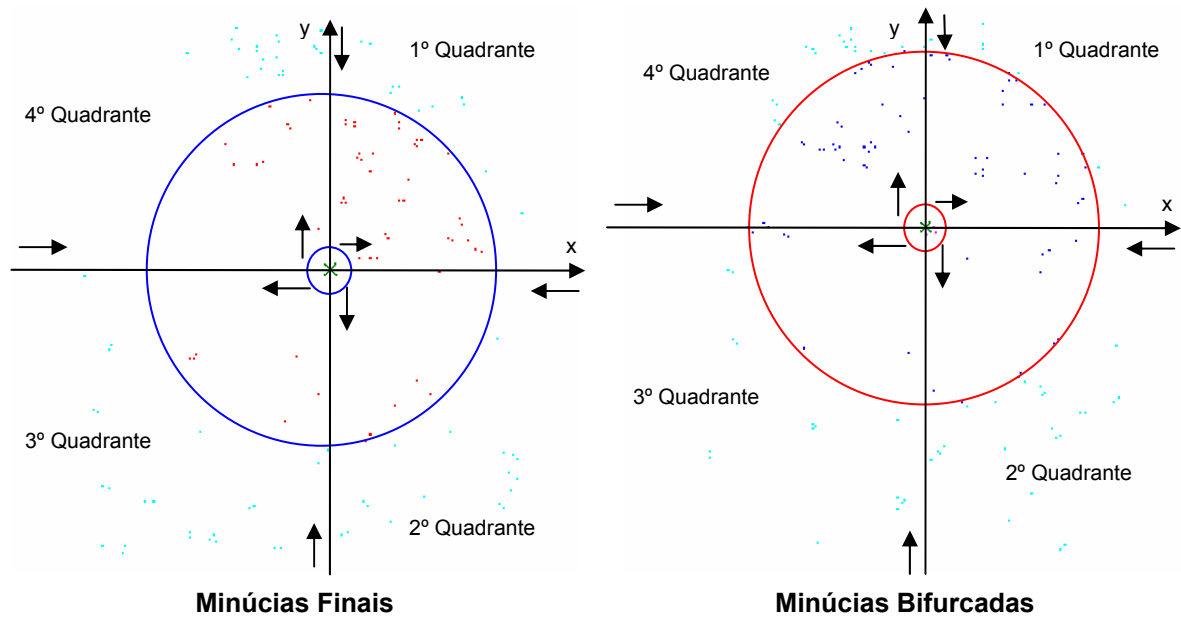


Figura 6.5: Minúcias Finais e Bifurcadas com a origem deslocada para o Centro de Massa da Imagem Original

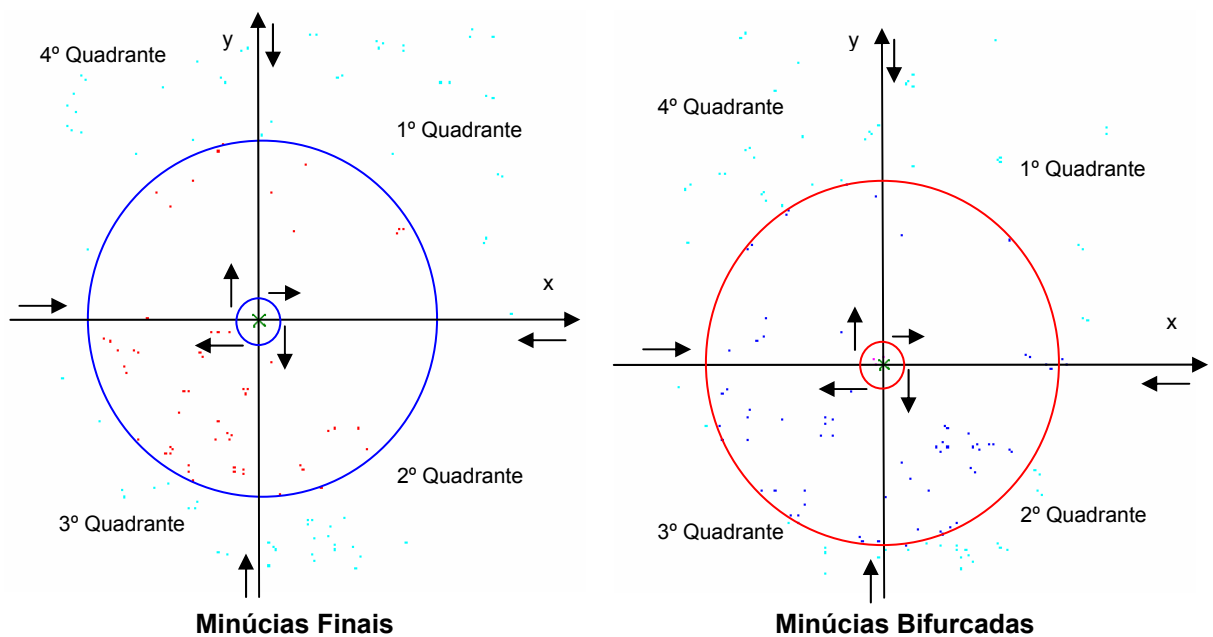


Figura 6.6: Minúcias Finais e Bifurcadas com a origem deslocada para o Centro de Massa da Imagem Rotacionada de 180º

Para realizar sucessivas rotações do Vetor Distância de Autenticação, determina-se um Vetor de Alinhamento para as minúcias finais e outro para as minúcias bifurcadas. Esta ordenação conterá quatro posições do Vetor Distância de Registro que será confrontada por sucessivas vezes com o Vetor Distância de Autenticação de quatro em quatro posições, calculando-se a distância entre elas,

localizando-se a menor distância que será o ponto de partida da ordenação do Vetor Distância de Autenticação. Assim, gera-se um ponto de partida para as minúcias finais e outro para as bifurcadas. A Tabela 6.1 mostra os Vetores Distância das minúcias finais para as duas imagens: um Vetor Distância de Registro (Imagem Original) e um Vetor Distância de Autenticação (Imagem Rotacionada de 180°), que serão utilizados como exemplo para demonstrar o Vetor Alinhamento.

Tabela 6.1: Vetor Distância de Registro e Vetor Distância de Autenticação das Minúcias Finais

Vetor Distância de Registro (Imagem Original) das Minúcias Finais Centro de Massa (X) = 119 Centro de Massa (Y) = 144							Vetor Distância de Autenticação (Imagem Rotacionada a 180°) das Minúcias Finais Centro de Massa (X) = 138 Centro de Massa (Y) = 113				
	Quad.	X	Y	D.E.	Média	Média		Quad.	X	Y	D.E.
1	1	175	139	56.223	67,7672	48,9347	1	1	168	108	30.414
2	1	158	129	41.785		49,0262	2	1	170	107	32.558
3	1	187	124	70.88		3	1	205	106	67.365
4	1	157	84	71.021		4	1	203	106	65.376
5	1	159	83	72.945		5	1	168	106	30.806
6	1	159	81	74.626		6	1	204	102	66.91
7	2	101	139	18.682		7	1	202	102	64.938
8	2	72	101	63.702		8	1	188	101	51.42
9	2	72	111	57.428		9	1	190	100	53.6
10	2	71	114	56.604		10	1	143	100	13.928
11	2	63	97	73.11		11	1	187	99	50.961
12	2	56	126	65.521		12	1	182	95	47.539
13	2	54	125	67.72		13	1	143	95	18.682
14	2	46	117	77.833		14	1	190	94	55.362
15	2	43	135	76.531		15	1	143	92	21.587
16	3	89	149	30.414		16	1	154	88	29.682
17	3	87	150	32.558		17	1	152	87	29.53
18	3	52	151	67.365		18	1	208	84	75.769
19	3	54	151	65.376		19	1	204	82	72.918
20	3	89	151	30.806		20	1	169	82	43.841
21	3	53	155	66.91		21	1	202	81	71.554
22	3	55	155	64.938		22	1	169	80	45.277
23	3	69	156	51.42		23	1	178	78	53.151
24	3	67	157	53.6		24	1	194	75	67.676
25	3	114	157	13.928		25	1	196	74	69.893
26	3	70	158	50.961		26	1	193	74	67.424
27	3	75	162	47.539		27	1	196	72	71.028
28	3	114	162	18.682		28	1	190	60	74.25
29	3	67	163	55.362		29	1	152	60	54.818
30	3	114	165	21.587		30	1	151	58	56.515
31	3	103	169	29.682		31	1	161	55	62.394
32	3	105	170	29.53		32	1	158	54	62.298
33	3	49	173	75.769		33	1	151	51	63.348
34	3	53	175	72.918		34	1	147	48	65.62
35	3	56	175	70.214		35	1	146	45	68.469
36	3	88	175	43.841		36	2	137	64	49.01
37	3	88	177	45.277		37	2	88	74	63.411
38	3	79	179	53.151		38	2	82	68	71.84
39	3	63	182	67.676		39	2	77	84	67.543
40	3	61	183	69.893		40	2	64	95	76.158
41	3	64	183	67.424		41	2	61	97	78.645
42	3	61	185	71.028		67,7672	42	3	82	118	56.223
43	3	67	197	74.25		72,368	43	3	70	133	70.88
44	3	105	197	54.818		59,3185	44	3	100	173	71.021
45	3	106	199	56.515		45	3	98	174	72.945
46	3	96	202	62.394		46	3	98	176	74.626
47	3	99	203	62.298		47	4	156	118	18.682
48	3	106	206	63.348		48	4	185	156	63.702

49	3	110	209	65.62	49	4	185	146	57.428
50	3	111	212	68.469	50	4	186	143	56.604
51	4	120	193	49.01	51	4	194	160	73.11
52	4	169	183	63.411	52	4	201	131	65.521
53	4	175	189	71.84		53	4	203	132	67.72
54	4	180	173	67.543		54	4	211	140	77.833
55	4	193	162	76.158		55	4	214	122	76.531
56	4	196	160	78.645						

Legenda: Quad. = Quadrante, C.M. = Centro de Massa, D.E. = Distância Euclidiana

Observa-se na tabela 6.1 que a minúcia espúria (representada em azul) dificulta a localização do ponto de partida, e este problema foi resolvido com as doze combinações de quatro posições do Vetor Alinhamento explicado no item 5.2.5 do capítulo 5.

Após a localização dos pontos de partida, ordena-se o Vetor Distância de Autenticação de cada tipo de minúcia, separando-se as minúcias espúrias finais e bifurcadas tanto do Vetor Distância de Registro como as do Vetor Distância de Autenticação em outros vetores, que serão chamados de Vetores Espúrios de Registro e Vetores Espúrios de Autenticação. Observa-se na Tabela 6.1 que a quantidade de minúcias entre os dois vetores não são iguais, portanto com a separação das minúcias espúrias essa quantidade se igualará.

A Tabela 6.2 mostra os Vetores Distâncias das minúcias finais para as duas imagens: um Vetor Distância de Registro (Imagem Original) e um Vetor Distância de Autenticação (Imagem Rotacionada de 180°), após a ordenação do Vetor Distância de Autenticação, utilizando o ponto de partida localizado pelo Vetor Alinhamento. Estes Vetores serão utilizados como exemplo para demonstrar explicar as minúcias espúrias extraídas e armazenadas no Vetor Espúrio de Registro e no Vetor Espúrio de Autenticação.

Tabela 6.2: Vetor Distância de Registro e Vetor Distância de Autenticação ordenado a partir do ponto de partida das Minúcias Finais

Vetor Distância de Registro (Imagem Original) das Minúcias Finais Centro de Massa (X) = 119 Centro de Massa (Y) = 144				
	Quad.	X	Y	D.E.
1	1	175	139	56.223
2	1	158	129	41.785
3	1	187	124	70.88
4	1	157	84	71.021
5	1	159	83	72.945
6	1	159	81	74.626
7	2	101	139	18.682
8	2	72	101	63.702
9	2	72	111	57.428
10	2	71	114	56.604
11	2	63	97	73.11
12	2	56	126	65.521

Vetor Distância de Autenticação (Imagem Rotacionada a 180°) das Minúcias Finais Centro de Massa (X) = 138 Centro de Massa (Y) = 113				
	Quad.	X	Y	D.E.
42	3	82	118	56.223
43	3	70	133	70.88
44	3	100	173	71.021
45	3	98	174	72.945
46	3	98	176	74.626
47	4	156	118	18.682
48	4	185	156	63.702
49	4	185	146	57.428
50	4	186	143	56.604
51	4	194	160	73.11
52	4	201	131	65.521

13	2	54	125	67.72
14	2	46	117	77.833
15	2	43	135	76.531
16	3	89	149	30.414
17	3	87	150	32.558
18	3	52	151	67.365
19	3	54	151	65.376
20	3	89	151	30.806
21	3	53	155	66.91
22	3	55	155	64.938
23	3	69	156	51.42
24	3	67	157	53.6
25	3	114	157	13.928
26	3	70	158	50.961
27	3	75	162	47.539
28	3	114	162	18.682
29	3	67	163	55.362
30	3	114	165	21.587
31	3	103	169	29.682
32	3	105	170	29.53
33	3	49	173	75.769
34	3	53	175	72.918
35	3	56	175	70.214
36	3	88	175	43.841
37	3	88	177	45.277
38	3	79	179	53.151
39	3	63	182	67.676
40	3	61	183	69.893
41	3	64	183	67.424
42	3	61	185	71.028
43	3	67	197	74.25
44	3	105	197	54.818
45	3	106	199	56.515
46	3	96	202	62.394
47	3	99	203	62.298
48	3	106	206	63.348
49	3	110	209	65.62
50	3	111	212	68.469
51	4	120	193	49.01
52	4	169	183	63.411
53	4	175	189	71.84
54	4	180	173	67.543
55	4	193	162	76.158
56	4	196	160	78.645

53	4	203	132	67.72
54	4	211	140	77.833
55	4	214	122	76.531
1	1	168	108	30.414
2	1	170	107	32.558
3	1	205	106	67.365
4	1	203	106	65.376
5	1	168	106	30.806
6	1	204	102	66.91
7	1	202	102	64.938
8	1	188	101	51.42
9	1	190	100	53.6
10	1	143	100	13.928
11	1	187	99	50.961
12	1	182	95	47.539
13	1	143	95	18.682
14	1	190	94	55.362
15	1	143	92	21.587
16	1	154	88	29.682
17	1	152	87	29.53
18	1	208	84	75.769
19	1	204	82	72.918
20	1	169	82	43.841
21	1	202	81	71.554
22	1	169	80	45.277
23	1	178	78	53.151
24	1	194	75	67.676
25	1	196	74	69.893
26	1	193	74	67.424
27	1	196	72	71.028
28	1	190	60	74.25
29	1	152	60	54.818
30	1	151	58	56.515
31	1	161	55	62.394
32	1	158	54	62.298
33	1	151	51	63.348
34	1	147	48	65.62
35	1	146	45	68.469
36	2	137	64	49.01
37	2	88	74	63.411
38	2	82	68	71.84
39	2	77	84	67.543
40	2	64	95	76.158
41	2	61	97	78.645

Legenda: Quad. = Quadrante, C.M. = Centro de Massa, D.E. = Distância Euclidiana

A Tabela 6.3 mostra os Vetores Espúrios das minúcias finais para as duas imagens: um Vetor Espúrio de Registro (Imagem Original) e um Vetor Espúrio de Autenticação (Imagem Rotacionada de 180°).

Tabela 6.3: Vetor Espúrio de Registro e Vetor Espúrio de Autenticação das Minúcias Finais

Vetor Espúrio de Registro (Imagem Original) das Minúcias Finais Centro de Massa (X) = 119 Centro de Massa (Y) = 144			
S.D.R.	D.E.	Quad.	S.D.A.
2	41.785	1	42

Vetor Espúrio de Autenticação (Imagem Rotacionada a 270°) das Minúcias Finais Centro de Massa (X) = 138 Centro de Massa (Y) = 113			
S.D.A.	D.E.	Quad.	S.D.R.
Nenhum			

Legenda: Quad. = Quadrante, C.M. = Centro de Massa, D.E. = Distância Euclidiana,
S.D.R. = Seqüência no Vetor Distância de Registro,
S.D.A. = Seqüência no Vetor Distância de Autenticação

A tabela 6.4 mostra o pior caso de Vetores Distâncias das minúcias finais para as duas imagens: um Vetor Distância de Registro (Imagem Original) e um Vetor Distância de Autenticação (Imagem Rotacionada de 270°) com várias minúcias

espúrias, dificultando a localização do Vetor Alinhamento. Estas tabelas serão utilizadas como exemplo para demonstrar que nos Vetores Espúrios de Registro e de Autenticação (Tabela 6.5) são encontradas minúcias em comuns, possibilitando aplicar a intuição humana citada por Dixon (1979) para reavaliar estas minúcias e re-incluí-las nos Vetores Distância.

Tabela 6.4: Pior caso de Vetor Distância de Registro e Vetor Distância de Autenticação das Minúcias Finais

Vetor Distância de Registro (Imagem Original) das Minúcias Finais Centro de Massa (X) = 143 Centro de Massa (Y) = 139				
	Quad.	X	Y	D.E.
1	1	187	134	44.283
2	1	185	133	42.426
3	1	196	113	59.034
4	1	168	112	36.797
5	1	168	108	39.825
6	1	200	106	65.863
7	1	192	106	59.076
8	1	195	104	62.682
9	1	169	103	44.407
10	1	168	100	46.325
11	1	203	95	74.404
12	1	198	89	74.33
13	1	163	82	60.407
14	1	163	80	62.298
15	1	163	77	65.146
16	2	94	78	78.243
17	3	80	140	63.008
18	3	117	142	26.173
19	3	131	144	13
20	3	133	144	11.18
21	3	106	147	37.855
22	3	118	149	26.926
23	3	133	155	18.868
24	3	78	158	67.72
25	3	78	160	68.308
26	3	122	161	30.414
27	3	140	166	27.166
28	3	78	171	72.45
29	3	89	181	68.411
30	4	151	210	71.449
31	4	153	207	68.731
32	4	153	201	62.801
33	4	153	199	60.828
34	4	153	165	27.857
35	4	154	215	76.792
36	4	157	166	30.414
37	4	162	163	30.61
38	4	162	160	28.32
39	4	173	189	58.31
40	4	175	210	77.878
41	4	176	190	60.745
42	4	179	210	79.605
43	4	179	190	62.426
44	4	193	185	67.941
45	4	201	161	62.032
46	4	211	166	73.164
47	4	211	144	68.184
48	4	213	144	70.178
49	4	214	142	71.063

Vetor Distância de Autenticação (Imagem Rotacionada a 270°) das Minúcias Finais Centro de Massa (X) = 138 Centro de Massa (Y) = 113				
	Quad.	X	Y	D.E.
21	2	137	34	79.006
22	2	134	70	43.186
23	2	133	72	41.304
24	2	121	35	79.831
25	2	113	61	57.697
26	2	112	89	35.384
27	2	108	89	38.419
28	2	106	57	64.498
29	2	106	65	57.689
30	2	104	62	61.294
31	2	103	88	43.012
32	2	100	89	44.944
33	2	95	54	73.007
34	2	89	59	72.918
35	2	88	52	78.873
36	2	82	94	59.135
37	2	80	94	61.033
38	2	77	94	63.891
39	3	78	163	78.102
40	4	139	123	10.05
41	4	140	177	64.031
42	4	142	140	27.295
43	4	144	126	14.318
44	4	144	124	12.53
45	4	147	151	39.051
46	4	149	139	28.231
47	4	155	124	20.248
48	4	158	179	68.964
49	4	160	179	69.57
50	4	161	135	31.828
51	4	166	117	28.284
52	4	171	179	73.79
53	4	181	168	69.814
1	1	210	106	72.339
2	1	208	105	70.456
3	1	207	104	69.584
4	1	201	104	63.64
5	1	199	104	61.66
6	1	165	104	28.46
7	1	215	103	77.647
8	1	166	100	30.871
9	1	163	95	30.806
10	1	160	95	28.425
11	1	189	84	58.669
12	1	210	82	78.39
13	1	190	81	61.057
14	1	190	78	62.682
15	1	185	64	67.897
16	1	161	56	61.465
17	1	166	46	72.615
18	1	144	46	67.268
19	1	144	44	69.26
20	1	142	43	70.114

Legenda: Quad. = Quadrante, C.M. = Centro de Massa, D.E. = Distância Euclidiana

Tabela 6.5: Vetor Espúrio de Registro e Vetor Espúrio de Autenticação das Minúcias Finais com minúcias espúrias em comum

</

Observando-se os Vetores Espúrios pode-se encontrar as três primeiras minúcias espúrias do Vetor Espúrio de Registro no Vetor Espúrio de Autenticação realizando sucessivas rotações, verificando se a diferença das Distâncias Euclidianas não ultrapassa o valor de limiar de 2.5 e a diferença entre as seqüências no Vetor Distância de Autenticação não ultrapassa o limiar de 5. Conclui-se que houve um pequeno deslocamento e que estas minúcias poderão ser re-incluídas nos Vetores Distâncias de Registro e de Autenticação, conforme demonstrado nas Tabelas 6.6 e 6.7.

Tabela 6.6: Vetor Distância de Registro e Vetor Distância de Autenticação das Minúcias Finais com minúcias espúrias re-incluídas

Vetor Distância de Registro (Imagem Original) das Minúcias Finais Centro de Massa (X) = 143 Centro de Massa (Y) = 139					Vetor Distância de Autenticação (Imagem Rotacionada a 270°) das Minúcias Finais Centro de Massa (X) = 138 Centro de Massa (Y) = 113				
	Quad.	X	Y	D.E.		Quad.	X	Y	D.E.
1	1	187	134	44.283	22	2	134	70	43.186
2	1	185	133	42.426	23	2	133	72	41.304
3	1	196	113	59.034	25	2	113	61	57.697
4	1	168	112	36.797	27	2	108	89	38.419
6	1	200	106	65.863	28	2	106	57	64.498
7	1	192	106	59.076	29	2	106	65	57.689
8	1	195	104	62.682	30	2	104	62	61.294
9	1	169	103	44.407	31	2	103	88	43.012
10	1	168	100	46.325	32	2	100	89	44.944
11	1	203	95	74.404	33	2	95	54	73.007
12	1	198	89	74.33	34	2	89	59	72.918
13	1	163	82	60.407	36	2	82	94	59.135
14	1	163	80	62.298	37	2	80	94	61.033
15	1	163	77	65.146	38	2	77	94	63.891
16	2	94	78	78.243	39	3	78	163	78.102
17	3	80	140	63.008	41	4	140	177	64.031
18	3	117	142	26.173	42	4	142	140	27.295
19	3	131	144	13	43	4	144	126	14.318
20	3	133	144	11.18	44	4	144	124	12.53
21	3	106	147	37.855	45	4	147	151	39.051
22	3	118	149	26.926	46	4	149	139	28.231
23	3	133	155	18.868	47	4	155	124	20.248
24	3	78	158	67.72	48	4	158	179	68.964
25	3	78	160	68.308	49	4	160	179	69.57

26	3	122	161	30.414
27	3	140	166	27.166
28	3	78	171	72.45
29	3	89	181	68.411
30	4	151	210	71.449
31	4	153	207	68.731
32	4	153	201	62.801
33	4	153	199	60.828
34	4	153	165	27.857
35	4	154	215	76.792
36	4	157	166	30.414
37	4	162	163	30.61
38	4	162	160	28.32
39	4	173	189	58.31
40	4	175	210	77.878
41	4	176	190	60.745
43	4	179	190	62.426
44	4	193	185	67.941
45	4	201	161	62.032
46	4	211	166	73.164
47	4	211	144	68.184
48	4	213	144	70.178
49	4	214	142	71.063

50	4	161	135	31.828
51	4	166	117	28.284
52	4	171	179	73.79
53	4	181	168	69.814
1	1	210	106	72.339
2	1	208	105	70.456
4	1	201	104	63.64
5	1	199	104	61.66
6	1	165	104	28.46
7	1	215	103	77.647
8	1	166	100	30.871
9	1	163	95	30.806
10	1	160	95	28.425
11	1	189	84	58.669
12	1	210	82	78.39
13	1	190	81	61.057
14	1	190	78	62.682
15	1	185	64	67.897
16	1	161	56	61.465
17	1	166	46	72.615
18	1	144	46	67.268
19	1	144	44	69.26
20	1	142	43	70.114

Legenda: Quad. = Quadrante, C.M. = Centro de Massa, D.E. = Distância Euclidiana

Tabela 6.7: Vetor Espúrio de Registro e Vetor Espúrio de Autenticação das Minúcias Finais após re-inclusão das minúcias espúrias em comum

Vetor Espúrio de Registro (Imagem Original) das Minúcias Finais Centro de Massa (X) = 143 Centro de Massa (Y) = 139			
S.D.R.	D.E.	Quad.	S.D.A.
5	39.825	1	28
42	79.605	4	14

Vetor Espúrio de Autenticação (Imagem Rotacionada a 270°) das Minúcias Finais Centro de Massa (X) = 138 Centro de Massa (Y) = 113			
S.D.A.	D.E.	Quad.	S.D.R.
26	35.384	2	3
35	78.873	2	14
40	10.05	4	18
3	69.584	1	33
21	79.006	2	49
24	79.831	2	49

Legenda: Quad. = Quadrante, C.M. = Centro de Massa, D.E. = Distância Euclidiana,
S.D.R. = Seqüência no Vetor Distância de Registro,
S.D.A. = Seqüência no Vetor Distância de Autenticação

Com estes vetores definidos e separados por tipo de minúcia, foram implementadas duas técnicas de Dixon (1979), tanto para as finais de linha do Vetor de Autenticação e Registro quanto para as bifurcadas do Vetor de Autenticação e de Registro. Os métodos foram citados no item 5.2.6 do capítulo 5. O método 2, será chamado de método Média e o método 3 de método Normal.

O método Média, substitui a minúcia espúria pela média entre os dois vizinhos mais próximos. Na tabela 6.7 referente ao Vetor Espúrio de Registro tem-se a primeira minúcia a ser substituída no Vetor Distância de Autenticação pela média dos dois vizinhos mais próximos da minúcia espúria. Observe-se que a minúcia espúria está localizada na seqüência 5 do Vetor Distância de Registro e os vizinhos da seqüência 5 são o 4 e 6, que possuem a Distância Euclidiana nos valores de (36.797, 39.825, 65.863) respectivamente; a média destes valores é 47.495. Finalmente, inclui-se no Vetor Distância de Autenticação na posição entre a

seqüência 27 e 28 a nova minúcia com valor 47.495. E assim, sucessivamente, até encerrar o Vetor Espúrio de Registro.

Após aplicar o método Média, calcula-se a Distância de Minkowsky entre o Vetor Distância de Registro e o Vetor Distância de Autenticação de cada imagem por tipo de minúcia. Sendo que duas imagens de impressão digital serão consideradas similares quando a Distância de Minkowsky entre elas for próxima de zero.

O Método Normal, utiliza o seguinte critério:

Calcula-se a Distância de Minkowsky entre o Vetor Distância de Registro (x_i) e o Vetor Distância de Autenticação (x_k) de cada imagem por tipo de minúcia sem se preocupar com as minúcias espúrias. Aplica-se posteriormente a seguinte equação:

$$Normal(i, k) = \frac{d}{d - d_o} \sum (d_i - d_k)^2 \quad (6.1)$$

Onde d_o é a quantidade de minúcias espúrias do Vetor Espúrio de Registro e d é a quantidade minúcias do Vetor Distância de Registro.

Para validação dos resultados do algoritmo proposto neste trabalho foi utilizado o método de Medidas Baseadas em Relevância, conhecido como *Recall x Precision* (JUNKER *et al*, 1999).

Para cada imagem rotacionada o método busca de 1 a 5 imagens, e armazena o valor do *Recall* e do *Precision*. O *Recall* armazena a porcentagem de imagens recuperadas da base de dados, e o *Precision* armazena a porcentagem de imagens recuperadas mas que sejam relevantes, ou seja, que faça parte do grupo de imagens verdadeiras.

Dixon (1979) afirma que o Método Normal tem performance superior ao Método Média. Relativamente a tempo de busca, nosso trabalho não permite comprovar esta afirmação devido a busca seqüencial implementada. Pode-se observar, no entanto, que o Método Normal apresenta desempenho superior ao Método da Média na recuperação das 672 imagens de impressões digitais rotacionadas.

A figura 6.7 mostra o Gráfico da Curva *Recall x Precision* da busca seqüencial exaustiva da base de imagens utilizando o método Média.

A figura 6.8 mostra o Gráfico da Curva *Recall x Precision* da busca seqüencial exaustiva da base de imagens utilizando o método Normal.

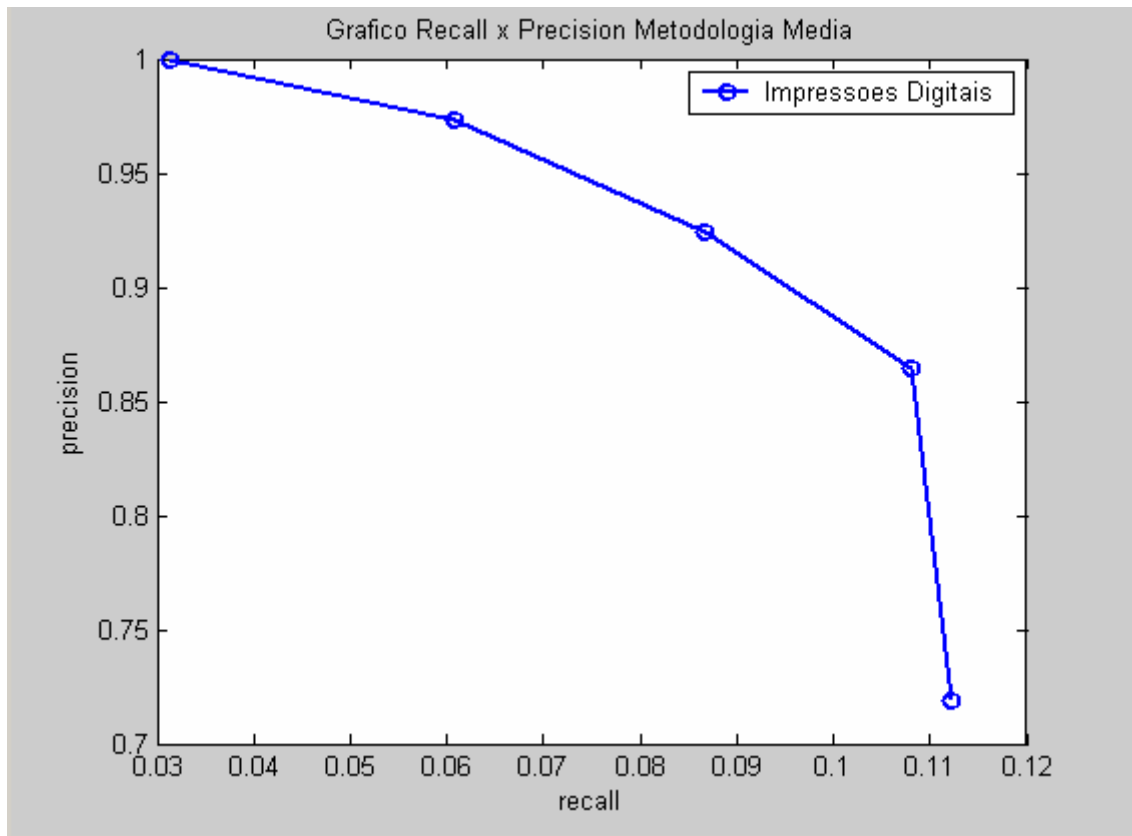


Figura 6.7: Gráfico Recall x Precision do Método Média

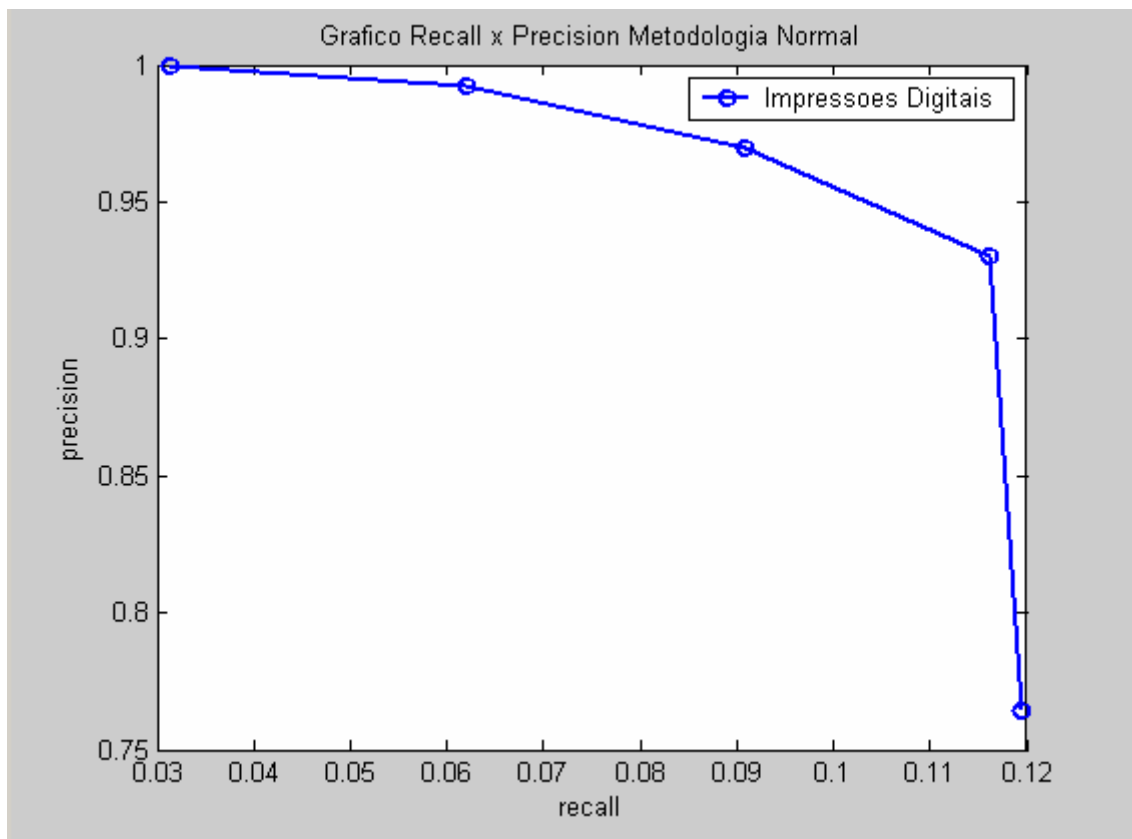


Figura 6.8: Gráfico Recall x Precision do Método Normal

As curvas de *Recall* x *Precision* (Figura 6.7 e 6.8) estimam a qualidade dos resultados do algoritmo após uma busca exaustiva da base de imagens. A Figura 6.9 mostra um caso de sucesso da 1ª a 5ª imagem mais próxima, buscando individualmente a imagem da pessoa número 1, quarta aquisição, rotacionada em 180°.

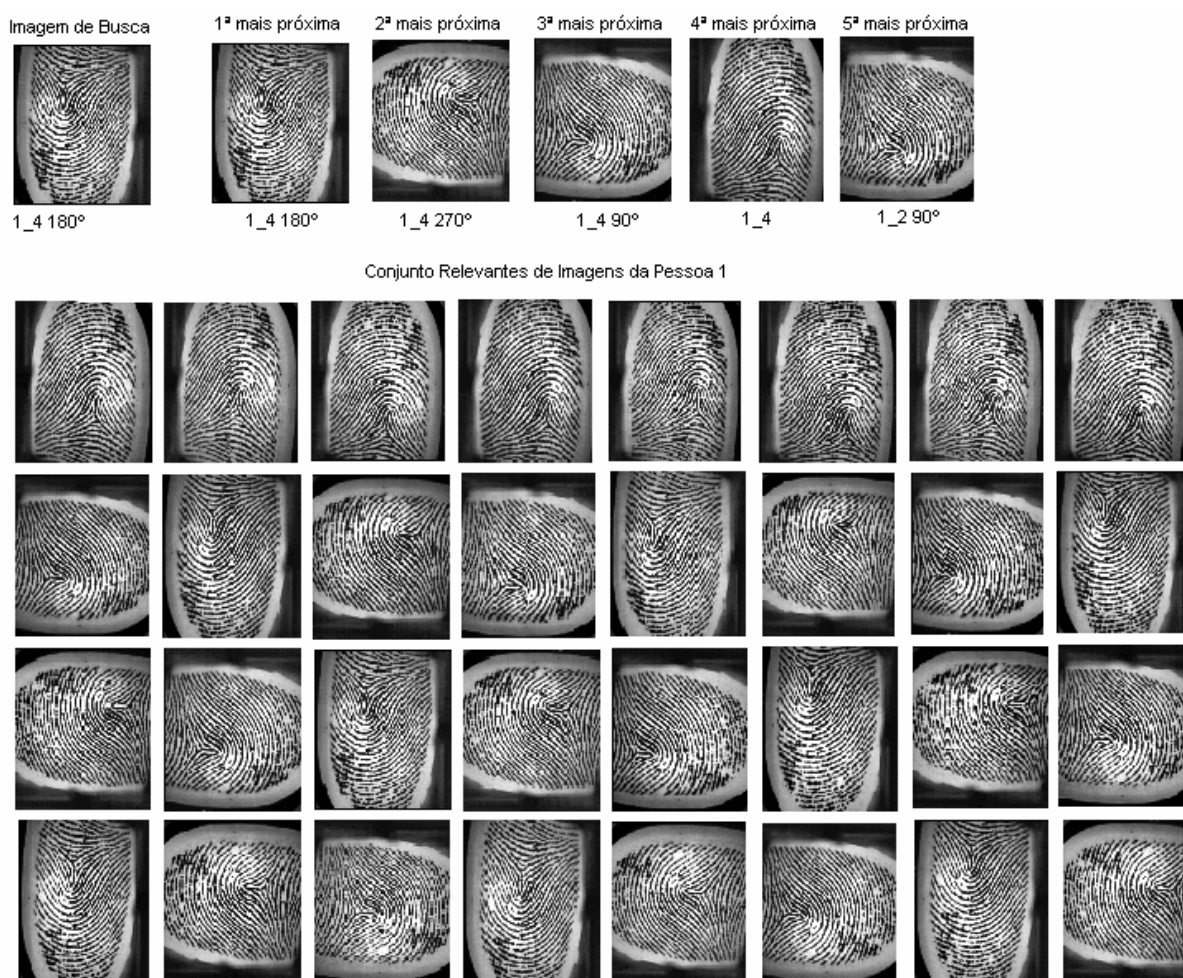


Figura 6.9: Caso de sucesso da 1ª a 5ª imagem mais próxima, buscando a imagem da pessoa número 1, quarta aquisição, rotacionada em 180°

A Figura 6.10 mostra um caso de insucesso na 2ª imagem mais próxima, buscando individualmente a imagem da pessoa número 10, quarta aquisição, rotacionada em 180°. O algoritmo localizou a imagem 7_5 em posição de origem. O caso de insucesso ocorreu pela baixa qualidade da imagem 10_4 (ver Figura 6.11) e pela baixa quantidade de pontos de minúcias encontradas pelo extrator de características, 8 pontos de minúcias do tipo final de linha e 9 para bifurcadas.

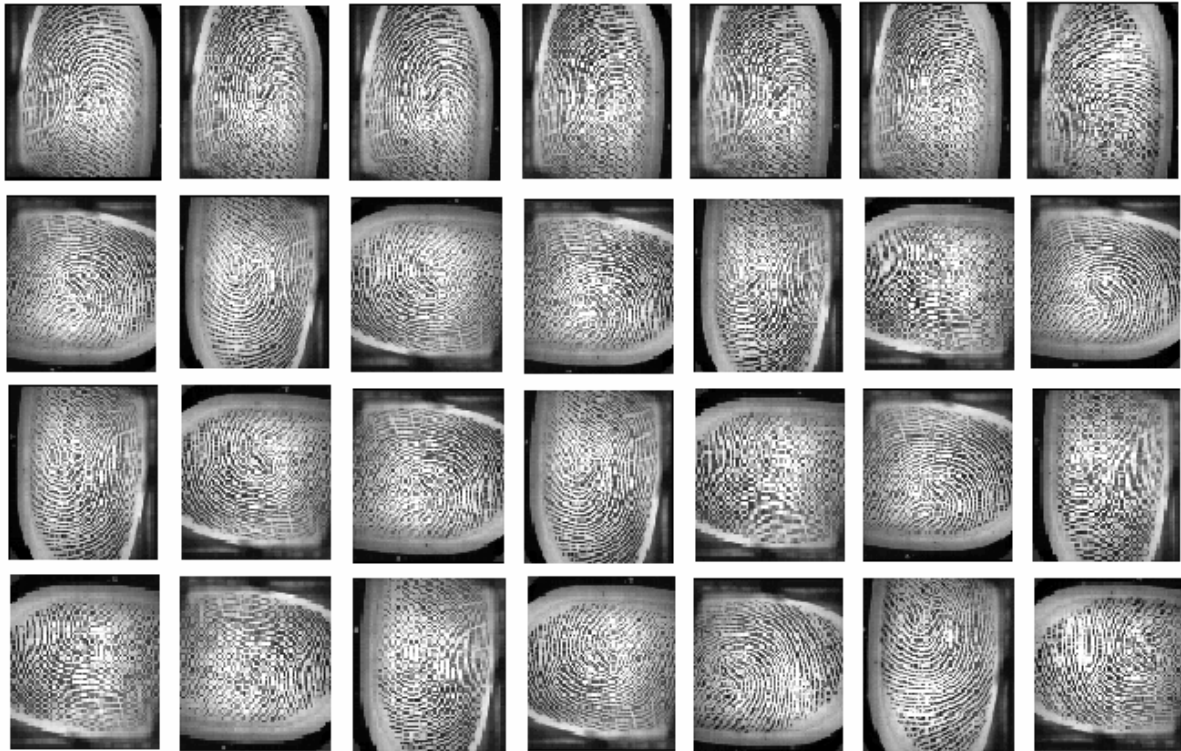


Figura 6.10: Caso de insucesso na 2ª imagem mais próxima, buscando a imagem da pessoa número 10, quarta aquisição, rotacionada em 180°

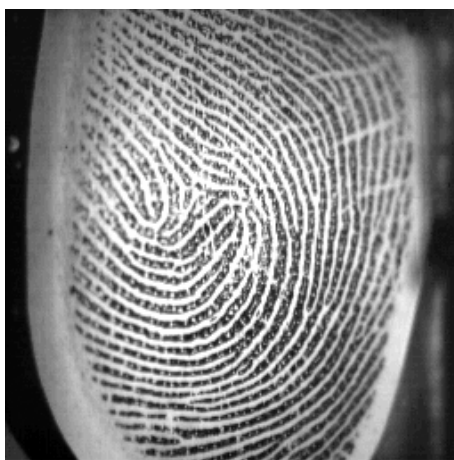


Imagem 10_4 Rotacionada 180°



Imagem 7_5

Figura 6.11: Imagens que ocasionaram o insucesso do algoritmo

6.2. Conclusões

Neste trabalho foi apresentada a proposta de um algoritmo para extração de características invariantes à rotação em imagens de impressões digitais. A etapa de pré-processamento da imagem, utiliza o detector de bordas de “*Canny*” e tem como resultado uma imagem binarizada e afinada. A etapa de extração de minúcias adota a Número de Cruzamentos como metodologia, e gera um vetor para as minúcias finais e outro para as minúcias bifurcadas. Considerando-se a posição (x,y) das minúcias, calcula-se o Centro de Massa desta distribuição e adota-se a origem do Sistema de Coordenadas sobre este Centro de Massa. Calcula-se a Distância Euclidiana dessa posição (x,y) ao Centro de Massa do vetor para cada tipo de minúcia (final e bifurcada), gerando-se Vetores Distâncias com seus elementos dispostos em ordem de quadrantes e no sentido horário, considerando as minúcias estáveis, que estão localizadas com valor da Distância Euclidiana ao Centro de Massa superior a 10 e inferior a 80. Esta restrição foi adotada devido ao fato observado que as minúcias próximas ao Centro de Massa sofrem distorções de posicionamento quando a rotação da imagem é realizada, pois os softwares utilizam o centro da imagem para rotacioná-la.

Essa metodologia permitiu avaliar a invariância com relação à rotação das imagens de Impressões Digitais. Aplicando-se um Vetor de Alinhamento para realizar rotações sucessivas, elemento a elemento, dos Vetores de Distâncias das impressões digitais rotacionadas foi possível ordenar as distâncias de minúcias do tipo final de linha e do tipo bifurcada que compõe o Vetor de Características invariante à rotação. Esta ordenação permite determinar a similaridade entre estas e todas as outras impressões não rotacionadas verificando-se o menor valor da Distância Euclidiana entre cada Vetor. As minúcias espúrias prejudicam a localização do ponto de partida da ordenação, por definir pontos de partida falsos. Visando melhorar a localização destas minúcias falsas ou espúrias foram definidas doze combinações de quatro posições do Vetor Distância da imagem origem para depois ser efetuado a comparação com o Vetor Distância da imagem rotacionada. Se após a rotação da imagem, não existissem os problemas de extração de minúcias espúrias, a metodologia atingiria 100% de precisão. Mas, na prática o problema existe e para garantir um melhor desempenho do algoritmo, foram

implementadas duas excelentes técnicas estatísticas sugeridas por Dixon (1979) para dados faltantes ou espúrios, tanto para minúcias finais de linha quanto para as bifurcadas.

Ordenou-se, então, o Vetor Distância da imagem rotacionada dos dois tipos de minúcias, separando-se as minúcias espúrias tanto da imagem original com as da imagem rotacionada, em dois novos vetores denominados de Vetores Espúrios Origem e Rotacionado, respectivamente. Em seguida, foram incluídas as minúcias espúrias em comuns nos Vetores de Autenticação e de Registro, por tipo de minúcia, nos respectivos Vetores Distâncias, aplicando-se a metodologia Média e Normal de Dixon (1979) para dados faltantes.

A metodologia foi validada através de curvas *Recall x Precision* aplicadas em toda a base de imagens sob teste formada por 672 imagens rotacionadas.

As características invariantes com relação à rotação estabelecidas por nosso algoritmo poderiam apresentar uma qualidade superior após uma busca exaustiva, se não fossem os problemas advindos de minúcias faltantes ou espúrias que aparecem durante o processo de rotação e que de certa forma simulam o efeito de imagens que são adquiridas na prática, de forma rotacionada. O problema com as minúcias espúrias extraídas foi solucionado pelo método estatístico de Dixon (1979) mostrando ainda, conforme esperado, que o Método Normal apresentou desempenho superior ao Método da Média.

Os resultados obtidos, conforme interpretação das curvas *Recall x Precision* comprovam a hipótese formulada inicialmente onde foi proposto lidar com imagens rotacionadas de impressões digitais e buscá-las em uma base de imagens a impressão mais similar. A similaridade obtida com rotações a 90, 180, 270°. aplicando-se a metodologia proposta neste trabalho demonstra a viabilidade de se implantar na prática um sistema de Autenticação por Impressão Digital, pois o mesmo mostrou-se robusto com relação à rotação.

Pela curva da Figura 6.8 pode-se ver que nossa metodologia apresenta uma precisão de aproximadamente 94% para recuperação das 4 imagens de cada classe, ou seja, imagem original e rotacionadas de 90, 180 e 270°, considerando-se toda a base de 672 imagens reais. Alguns problemas ocorreram devido à má

qualidade das imagens, fato este normal em sistemas reais, mas que reduziram de uma parcela muito pequena a precisão global da metodologia proposta.

6.3. Perspectivas Futuras

Neste trabalho para efetuar a busca exaustiva da base de imagens de 672 imagens de impressões digitais, foi utilizada uma busca seqüencial, causando um custo computacional alto. Como proposta de trabalhos futuros propõe-se o uso de metodologias de indexação por conteúdo em um Sistema de Banco de Dados, como por exemplo a Slim-Tree, que certamente otimizará o tempo de busca em um conjunto maior de imagens do que as utilizadas neste trabalho para validar a metodologia.

Devido ao fato de se trabalhar com imagens previamente adquiridas, também utilizadas por outros pesquisadores, na ausência de um mecanismo de digitalização, estas imagens utilizadas tiveram que ser rotacionadas, causando distorções. Como os resultados obtiveram sucesso nestas condições, sugere-se que o algoritmo desenvolvido neste trabalho seja testado com imagens adquiridas em qualquer ângulo de rotação.

O pré-processamento é uma das etapas fundamentais para o tratamento das imagens, sugere-se a aplicação de outros detectores de bordas que utilizam metodologias diferentes do Detector de “*Canny*” utilizado neste trabalho.

O método utilizado para extração das minúcias foi o Número de Cruzamentos (CN) (AMENGUAL *apud* THAI, 2003), considerando os aspectos estruturais mais comuns, cristas finais e cristas bifurcadas. Podendo ser utilizados outros aspectos como os pontos isolados, os cruzamentos de pontos e as linhas contínuas.

É importante salientar que as rotações tratadas pela maioria dos trabalhos encontrados na literatura não excedem aos 30° o que dificulta a comparação de nossos resultados (rotações de qualquer ângulo) com qualquer outro trabalho publicado anteriormente.

Referências Bibliográficas

- ADHAMI, R.; MEENEN, P. Fingerprint for security. *IEEE Potentials*, p.33-38, 2001.
- AMENGUAL, J. C.; JUAN, A.; PREZ, J. C.; PRAT, F.; SEZ, S.; VILAR, J. M. Real-time minutiae extraction in fingerprint images. In: *Proc. of the 6th Int. Conf. on Image Processing and its Applications (July 1997)*, pp. 871–875 *apud* THAI, Raymond. *Fingerprint Image Enhancement and Minutiae Extraction*. School of Computer Science and Software Engineering, The University of Western Australia, 2003. Disponível em: <www.csse.uwa.edu.au/~pk/studentprojects/raymondthai/RaymondThai.pdf>. Acesso em: 28 dez. 2004.
- BOLLE, R.M. *et al. Biometrics Technologies ... Emerging into the Mainstream*. IBM Research Division, oct. 2001.
- CAPPELLI, R. *et al.* Fingerprint Classification by Directional Image Partitioning, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v.21, n. 5, pp.402-421, may 1999.
- CHANG, Jeng-Horng.; FAN, Kuo-Chin. A new model for fingerprint classification by ridge distribution sequences. *Pattern Recognition Society*, v. 35, pp.129-132, 2002.
- CHELLAPPA, R.; WILSON, C.;SIROHEY, A. Human and machine recognition of faces. A survey. *Proceedings IEEE*, 83(5):705-740, 1995 *apud* JAIN, A.K. *et al.* Biometrics: Promising Frontiers for Emerging Identification Market. *Communications of the ACM*, v. 43, n. 2, p.91-98, feb. 2000.
- COSTA, Silvia Maria Farani. *Classificação e verificação de Impressões Digitais*. 2001. Dissertação de Mestrado em Engenharia Elétrica - Escola Politécnica da Universidade de São Paulo, São Paulo, 2001. Disponível em: <http://www.teses.usp.br/teses/disponiveis/3/3140/tde-18032002-102113/publico/dissertacao_Silvia.pdf>. Acesso em: 20 dez.2004.
- ELECCION, Marce. *Pattern Recognition – Automatic Fingerprint Identification*. *IEEE Spectrum*, pp. 36-45, 1973.
- DAUGHMAN, J.G. High confidence visual recognition of persons by a test of statistical independence. *IEEE Trans.Pattern Anal. and Machine Intell*, 15(11):1148-1161, 1993 *apud* JAIN, A.K. *et al.* Biometrics: Promising Frontiers for Emerging Identification Market. *Communications of the ACM*, v. 43, n. 2, p.91-98, feb. 2000.
- DIXON, J.K. Pattern Recognition with Partly Missing Data. *IEEE Transactions on Systems, Man and Cybernetics*, v. 9, n.10, pp. 617-621, oct. 1979.

- FARINA, A.; KOVÁCS-VAJNA, Z.M.; LEONE, A. Fingerprint minutiae extraction from skeletonized binary images. *Pattern Recognition*, v. 32, pp. 877-889, 1999.
- GALTON, F. *"Fingerprints"*. Mcmillan, 1982.
- GALTON, F. *Fingerprints*. London: McMillan, 1892. *apud* CAPPELLI, R. *et al.* Fingerprint Classification by Directional Image Partitioning, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v.21, n. 5, pp.402-421, may 1999.
- GONZALEZ, R.C.; WOODS, R.E. *Processamento de Imagens Digitais*. Editora Edgard Blücher Ltda, 2003, p.509.
- GREENBERG, Shlomo; ALADJEM, Mayer; KOGAN, Daniel. Fingerprint Image Enhancement Using Filtering Techniques. *Real-Time Imaging*, n. 8, pp. 227–236, 2002
- HENRY, E.R. Classification and Uses of Finger Prints. London: Routledge, 1900. *apud* CAPPELLI, R. *et al.* Fingerprint Classification by Directional Image Partitioning, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v.21, n. 5, pp. 402-421, may 1999.
- HONG, L.; JAIN, A.K. Fingerprint Image Enhancement: Algorithm and Performance Evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v.20, n. 8, pp. 777-789, 1998.
- HONG, L.; JAIN, A.K. Integrating Faces and Fingerprints for Personal Identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 20, n. 12, pp. 1295-1307, dec. 1998.
- HRECHAK, A.K.; MCHUGH, J.A. *Automated Fingerprint Recognition using Structural Matching*. *Pattern Recognition*, v. 23, n. 8, pp. 893-904, 1990.
- JAIN, A.K.; HONG, L.; BOLLE, R. On-Line Fingerprint Verification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 19, n. 4, pp. 302-313, 1997.
- JAIN, A.K.; HONG, L.; PANKANTI, S. Biometrics: Promising Frontiers for Emerging Identification Market. *Communications of the ACM*, v. 43, n. 2, p.91-98, feb. 2000.
- JAIN, A.K.; PANKANTI, S. Fingerprint classification and matching. In: Bovik A., (ed.), *Handbook for Image and Video Processing*. New York : Academic Press. april, 2000b.
- JAIN, A.K.; PRABHAKAR, S.; PANKANTI, S. On the similarity of identical twin fingerprints. *Pattern Recognition Society*, v. 35, pp. 2653-2663, 2002.

- JUNKER, Markus; DENGEL, Andréas; HOCH, Rainer. On the Evaluation of Document Analysis Components by Recall, Precision, and Accuracy. *IEEE Computer Society. Fifth International Conference on Document Analysis and Recognition*, 1999, p. 713.
- KARU, Kalle; JAIN, Anil K. Fingerprint Classification. *Pattern Recognition Society*, v.29, n. 3, pp. 389-404, 1996.
- LINDLEY, C. A. *Practical Image Processing in C (Aquisition, Manipulation, Storage)*. New York : Wiley, 1991.
- LIU, S.; SILVERMAN, M. A practical guide to biometric security technology. *IEEE Computer Society IT Pro – Security*, jan-feb 2001 *apud* BOLLE, R.M. *et al. Biometrics Technologies ... Emerging into the Mainstream*. IBM Research Division. oct. 2001.
- LUMINI, A.; MAIO, D.; MALTONI, D., Continuous vs Exclusive Classification for Fingerprint Retrieval, *Pattern Recognition Letters*, v. 18, n. 10, pp. 1027-1034, 1997.
- MAIO, Dario; MALTONI, Davide. A Structural Approach to Fingerprint Classification. *IEEE Proceedings of ICPR'96*, pp. 578-585, 1996.
- MALTONI, D; MAIO, D.; JAIN, A.K.; PRABHAKAR, S., *Handbook of Fingerprint Recognition*. New York: Springer-Verlag, 2003, p. 309.
- MOAYER, B.; FU, K.S. A syntatic approach to fingerprint pattern recognition. *Pattern Recognition*, v. 7, pp.1-23, 1975 *apud* CHANG, Jeng-Horng.; FAN, Kuo-Chin. A new model for fingerprint classification by ridge distribution sequences. *Pattern Recognition Society*, v. 35, pp. 129-132, 2002.
- RATHA, Nalini K.; CHEN, Shaoyun; JAIN, Anil K. Adaptative Flow Orientation – Based Feature Extraction in Fingerprint Images. *Pattern Recognition*, v. 28, n.11, pp. 1657-1672, 1995.
- RATHA, Nalini; KARU, Kalle; CHEN, Shayun; JAIN, Anil K. A Real Time Matching System for Large Fingerprint Databases. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 18, n. 8, pp. 799-813, 1996.
- RATHA, N.K. *et al.* Enhancing security and privacy in biometrics-based authentication systems. *IBM Systems Journal*, v. 40, n. 3, pp. 614-634, 2001.
- SANDERSON, S.; ERBETTA, J.H. Authentication for secure environments based on iris scanning technology. Visual Biometrics (ref.nº 2000/018), *IEE Colloquium On*, pp.8/1-8/7, 2000.
- SENIOR, A.W. *et al. Fingerprint minutiae: a Constructive Definition*. Workshop on Biometrics, To appear in Copenhagen, Denmark June 2002, IEEE ECCV.

- TEDESCO, Claudia; COUTO, Daniel Lucena. *Desenvolvimento de Algoritmos para Análise de Imagens de Impressões Digitais Rotacionadas utilizando Grafos*. In: Faculdade Ruy Barbosa (FRB), [2001]. Disponível em: <www.uefs.br/erbase2004/documentos/wticgbase/Wticgbase2004ArtigoMonografia002.pdf>. Acesso em: 05 jan. 2005.
- THAI, Raymond. *Fingerprint Image Enhancement and Minutiae Extraction*. In: School of Computer Science and Software Engineering, The University of Western Australia, 2003. Disponível em: <www.csse.uwa.edu.au/~pk/studentprojects/raymondthai/RaymondThai.pdf>. Acesso em: 28 dez. 2004.

Apêndice A

Regras para a Direção de Crista Local Bifurcada

Para $\theta=0^\circ$, a crista que passa pelo *pixel* P que está sendo processado, deve passar obrigatoriamente pelo *pixel* P_1 e o par de *pixels* representando a bifurcação pode passar pelos outros *pixels* exceto P_2 e P_8 , e o valor de CN deve ser igual a 3. Exemplos de todas as possibilidades para $\theta=0^\circ$ são ilustrados na Figura A.1.

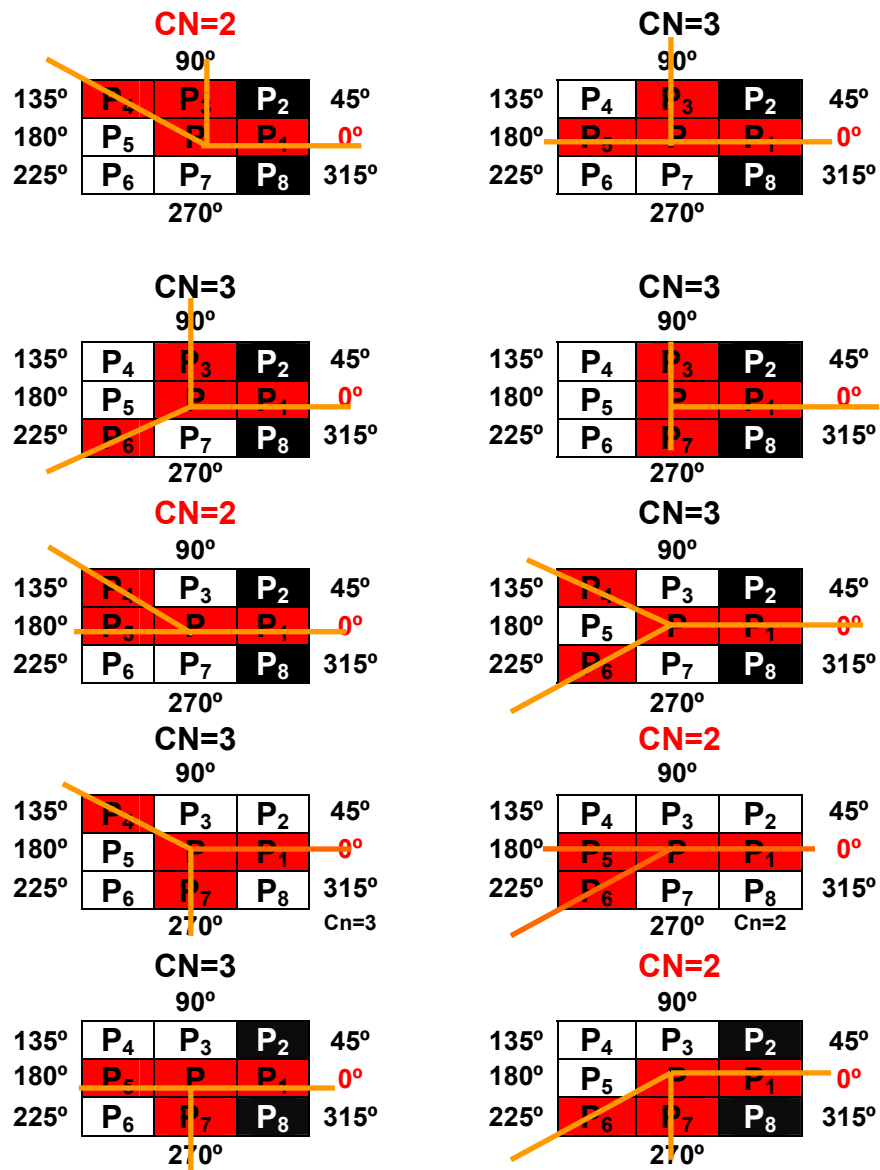


Figura A.1: Exemplos da Direção de Crista Local Bifurcada onde $\theta=0^\circ$

Para $\theta=45^\circ$, a crista que passa pelo *pixel* P que está sendo processado, deve passar obrigatoriamente pelo *pixel* P_2 e o par de *pixels* representando a bifurcação pode passar pelos outros *pixels* exceto P_1 , P_8 , P_3 e P_4 , e o valor de CN deve ser igual a 3. Exemplos de todas as possibilidades para $\theta=45^\circ$ são ilustrados na Figura A.2.

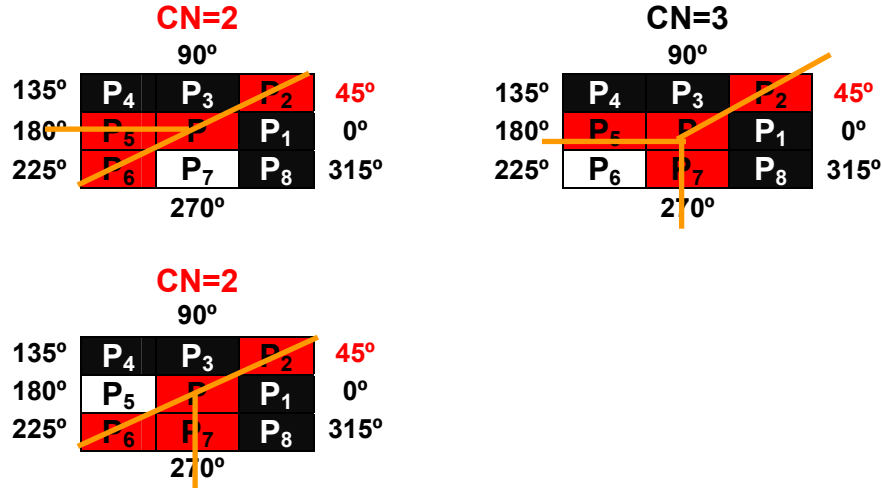
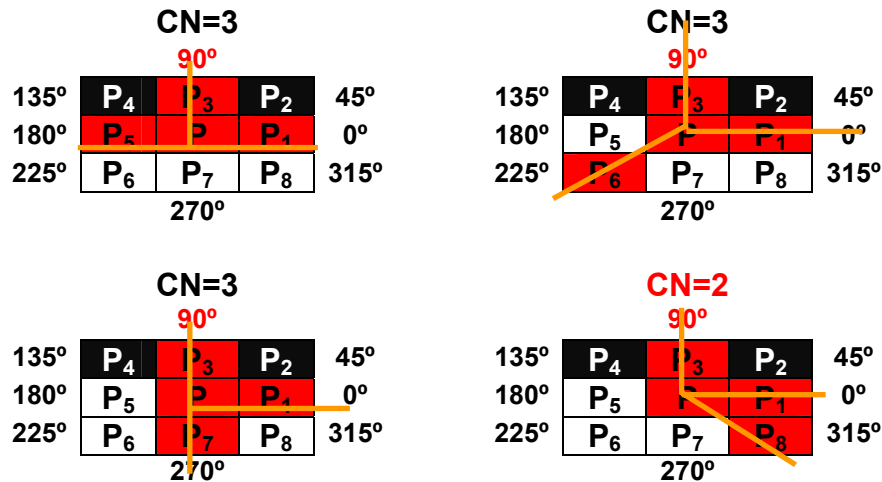


Figura A.2: Exemplos da Direção de Crista Local Bifurcada onde $\theta=45^\circ$

Para $\theta=90^\circ$, a crista que passa pelo *pixel* P que está sendo processado, deve passar obrigatoriamente pelo *pixel* P_3 e o par de *pixels* representando a bifurcação pode passar pelos outros *pixels* exceto P_2 e P_4 , e o valor de CN deve ser igual a 3. Exemplos de todas as possibilidades para $\theta=90^\circ$ são ilustrados na Figura A.3.



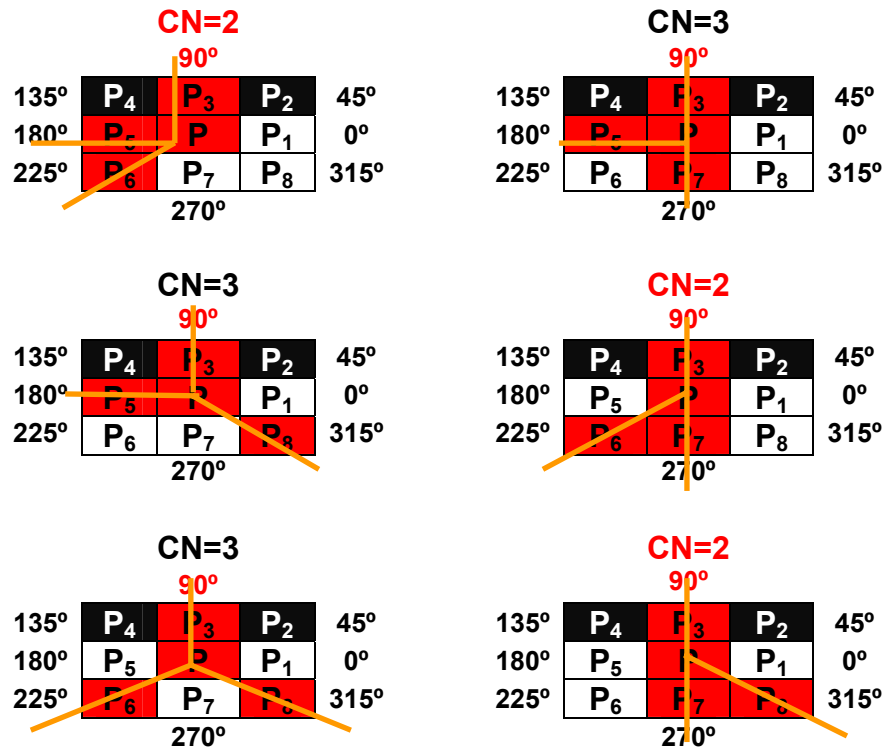


Figura A.3: Exemplos da Direção de Crista Local Bifurcada onde $\theta = 90^\circ$

Para $\theta = 135^\circ$, a crista que passa pelo *pixel* que está sendo processado, deve passar obrigatoriamente pelo *pixel* P_4 e o par de *pixels* representando a bifurcação pode passar pelos outros *pixels* exceto P_2 , P_3 , P_5 e P_6 , e o valor de CN deve ser igual a 3. Exemplos de todas as possibilidades para $\theta = 135^\circ$ são ilustrados na Figura A.4.

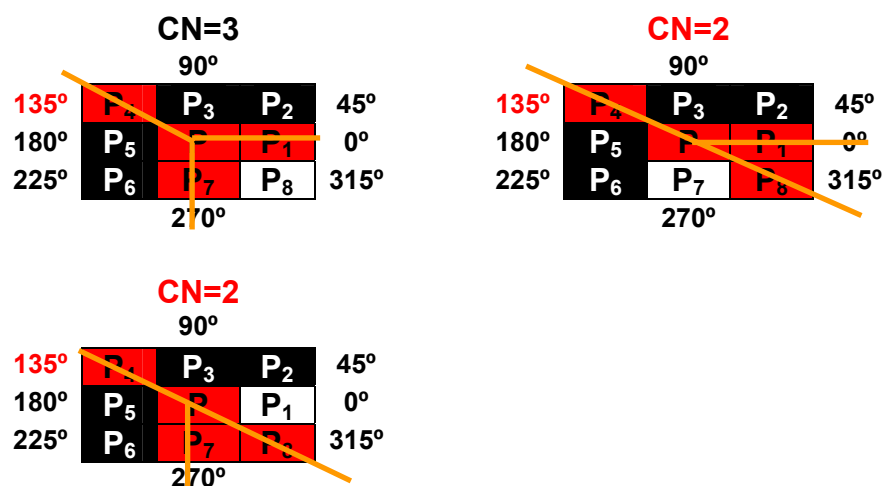


Figura A.4: Exemplos da Direção de Crista Local Bifurcada onde $\theta = 135^\circ$

Para $\theta=180^\circ$, a crista que passa pelo *pixel* P que está sendo processado, deve passar obrigatoriamente pelo *pixel* P_5 e o par de *pixels* representando a bifurcação pode passar pelos outros *pixels* exceto P_4 e P_6 , e o valor de CN deve ser igual a 3. Exemplos de todas as possibilidades para $\theta=180^\circ$ são ilustrados na Figura A.5.

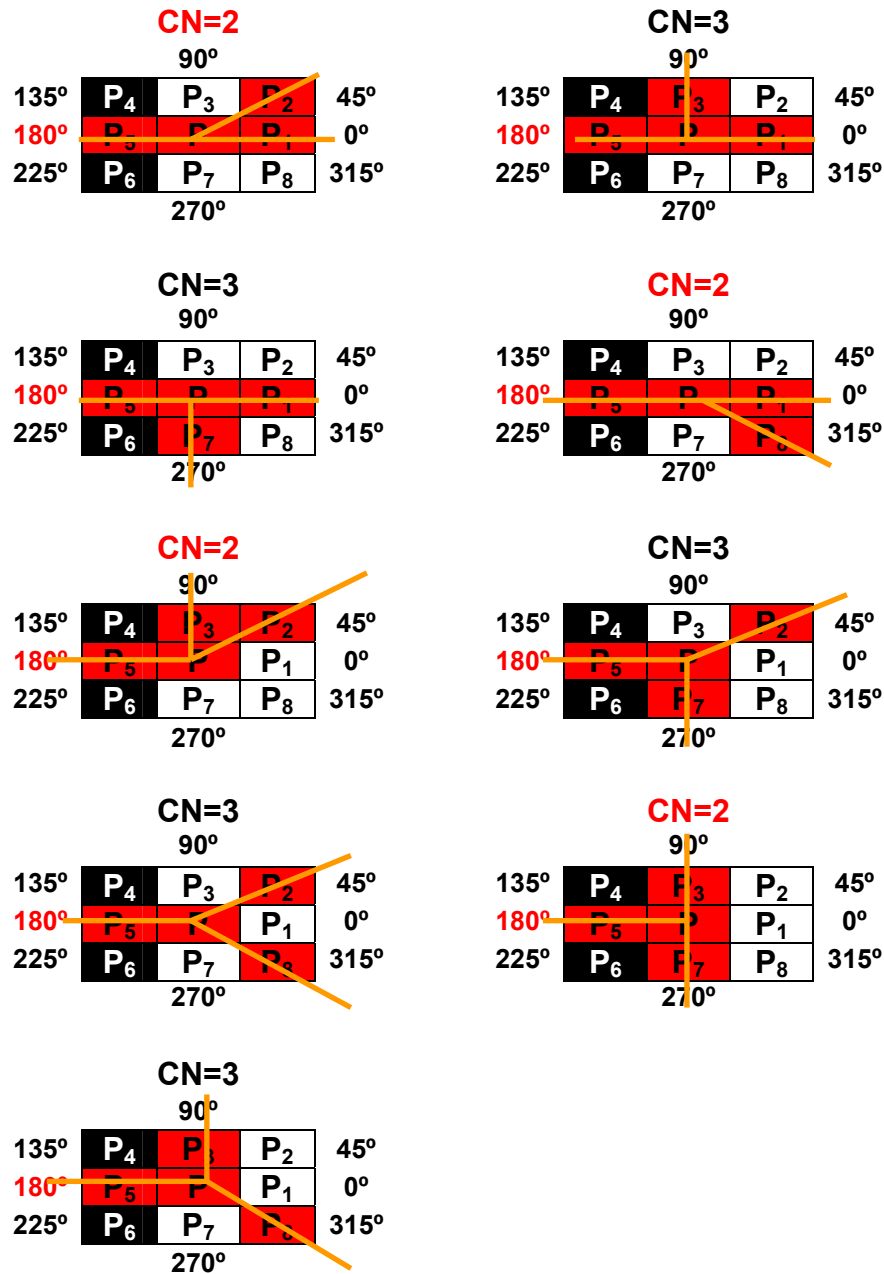


Figura A.5: Exemplos da Direção de Crista Local Bifurcada onde $\theta=180^\circ$

Para $\theta=225^\circ$, a crista que passa pelo *pixel* P que está sendo processado, deve passar obrigatoriamente pelo *pixel* P_6 e o par de *pixels* representando a bifurcação pode passar pelos outros *pixels* exceto P_4 , P_5 , P_7 e P_8 , e o valor de CN deve ser igual a 3. Exemplos de todas as possibilidades para $\theta=225^\circ$ são ilustrados na Figura A.6.

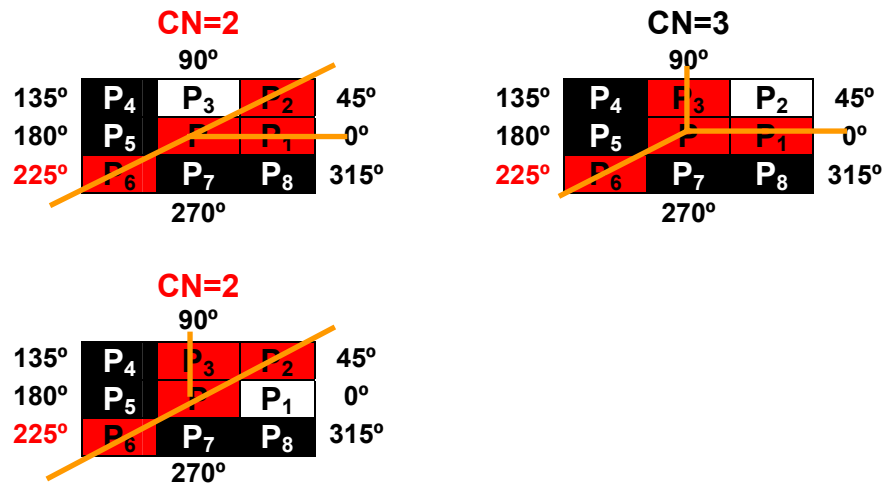
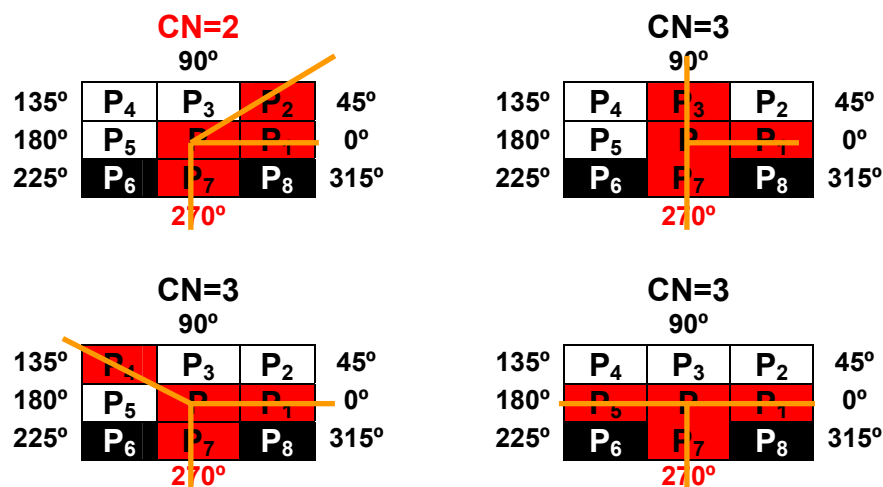


Figura A.6: Exemplos da Direção de Crista Local Bifurcada onde $\theta=225^\circ$

Para $\theta=270^\circ$, a crista que passa pelo *pixel* P que está sendo processado, deve passar obrigatoriamente pelo *pixel* P_7 e o par de *pixels* representando a bifurcação pode passar pelos outros *pixels* exceto P_6 e P_8 , e o valor de CN deve ser igual a 3. Exemplos de todas as possibilidades para $\theta=270^\circ$ são ilustrados na Figura A.7.



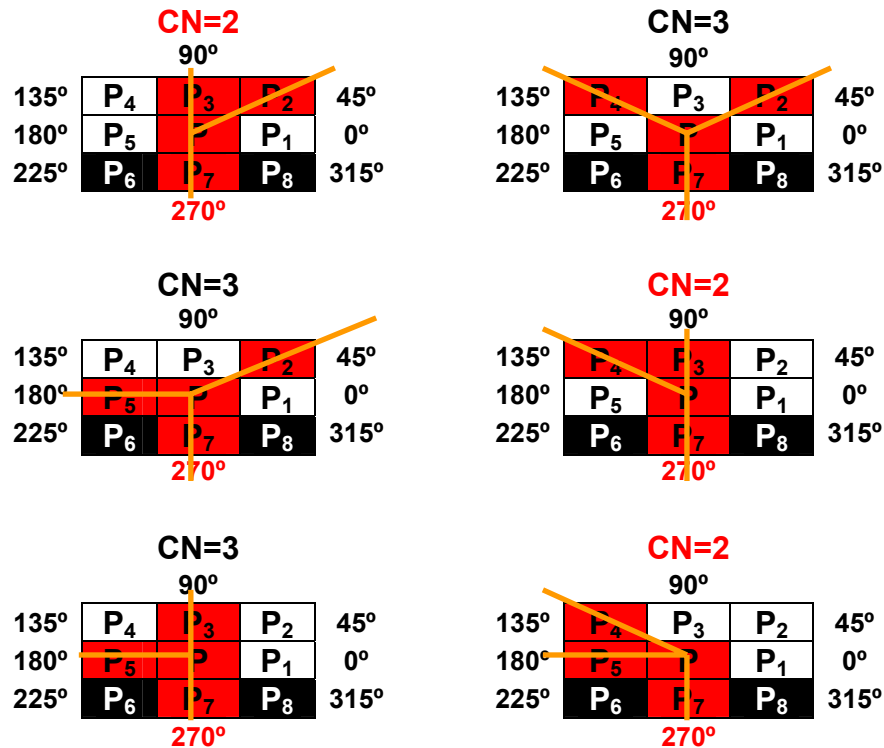


Figura A.7: Exemplos da Direção de Crista Local Bifurcada onde $\theta=270^\circ$

Para $\theta=315^\circ$, a crista que passa pelo *pixel* P que está sendo processado, deve passar obrigatoriamente pelo *pixel* P_8 e o par de *pixels* representando a bifurcação pode passar pelos outros *pixels* exceto P_2 , P_1 , P_7 e P_6 , e o valor de CN deve ser igual a 3. Exemplos de todas as possibilidades para $\theta=315^\circ$ são ilustrados na Figura A.8.

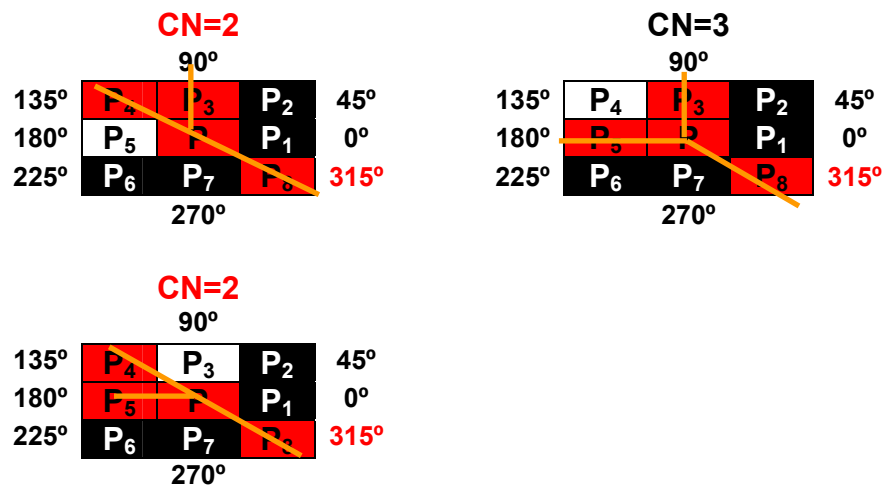


Figura A.8: Exemplos da Direção de Crista Local Bifurcada onde $\theta=315^\circ$

Apêndice B

O algoritmo foi desenvolvido com auxílio do *MatLab*. Esta ferramenta apresenta um *toolbox* para processamento de imagens com algumas funções pré-definidas, que facilitam o trabalho com imagens uma vez que é uma linguagem orientada ao tratamento de vetores e matrizes.

Lista de Programas

Todas as funções devem ser utilizadas para as imagens originais e rotacionadas.

- 1) **localizaMinucias:** Localiza as minúcias da impressão digital, calcula o Centro de Massa, separa as minúcias por quadrantes e calcula a distância do Centro de Massa a cada minúcia extraída ignorando as distâncias inferiores a 10 e superiores a 80. O algoritmo de localização das minúcias da impressão digital são de Thai (2003).
- 2) **ordenarVetorRotacionado:** Localiza o ponto inicial para realizar sucessivas rotações do Vetor Distância de Autenticação.
Esta função utiliza duas sub-funções: *vetorAlinhamento* e *verificaEspurio*.
A função *vetorAlinhamento* utiliza as sub-funções: *vetorAlinhamentoMedia 4Pontos*, *verificaIndexEncontrado* e *refazVetorAlinhamento*.
- 3) **reordenarVetorRotacionado:** Re-inclui as minúcias espúrios em comuns no Vetor Espúrio de Registro e Vetor Espúrio de Autenticação
- 4) **DixonMedia:** Para as minúcias espúrias extraídas armazenadas no Vetor Espúrio de Registro, substitui no Vetor Distância de Autenticação pela média dos dois vizinhos mais próximos.
- 5) **DistanciaEuclidianaMedia:** Para calcular a distância de Minkowski utilizando o Método Média de Dixon (1979).
- 6) **DistanciaEuclidianaNormal:** Para calcular a distância de Minkowski utilizando o Método Normal de Dixon (1979).

1) localizaMinucias

```
function [distanciaFim, distanciaBif] = localizaMinucias(img)

% convert to gray scale and thin image at same time
thinned = edge(RGB2GRAY(img), 'canny');

% minutiae counter
minu_count = 1;
minutiae(minu_count, :) = [0,0,0,0];
% loop through image and find minutiae, ignore 10 pixels for border
for x=10:size(img, 2) - 10
    for y=10:size(img, 1) - 10
        if (thinned(x, y) == 1) % only continue if pixel is white
            % calculate CN from Raymond Thai
            CN = 0;
            for i = 1:8
                CN = CN + abs (P(thinned, x, y, i) - P(thinned, x, y, i + 1));
            end
            CN = CN / 2;

            if (CN == 1) | (CN == 3)
                theta = FindTheta(thinned, x, y, CN);
                minutiae(minu_count, :) = [x, y, CN, theta];
                minu_count = minu_count + 1;
            end
        end % if pixel white
    end % for y
end % for x

% make minutiae image
%atualiza vetor com minucias final de linha separado do vetor com minucias de
%bifurcacao
minutiae_img = uint8(zeros(size(img, 1),size(img, 2), 3));
minutiae_fim = uint8(zeros(size(img, 1),size(img, 2), 3));
minutiae_bif = uint8(zeros(size(img, 1),size(img, 2), 3));
%contador para o ponto de fim de linha
linha_fim=1;
%contador para o ponto de bifurcacao
linha_bif=1;
%vetor para minucias fim de linha
minucias_fim(linha_fim, :) = [0,0,0];
%vetor para minucias bifurcações
minucias_bif(linha_bif, :) = [0,0,0];

for i=1:minu_count - 1
    x = minutiae(i, 1);
    y = minutiae(i, 2);
    theta = minutiae(i, 4);
    if (minutiae(i, 3) == 1)
        minucias_fim(linha_fim, :) = [x, y, theta];
        minutiae_fim(x, y,:) = [255, 0, 0]; % red for ridge endings
        minutiae_img(x, y,:) = [255, 0, 0];
        linha_fim = linha_fim + 1;
    else
        minucias_bif(linha_bif, :) = [x, y, theta];
        minutiae_bif(x, y,:) = [0, 0, 255]; % blue for ridge bijections
        minutiae_img(x, y,:) = [0, 0, 255];
    end
end
```

```

        linha_bif = linha_bif + 1;
    end
end % for loop through minu_count

% junta a imagem pre-processada (thinned) com a imagem das minucias (pontos fim (vermelho) e
% pontos bifurcacao (azul))
combined = uint8(minutiae_img);
for x=1:size(thinned)
    for y=1:size(thinned)
        if (thinned(x,y))
            combined(x,y,:) = [255,255,255];
        else
            combined(x,y,:) = [0,0,0];
        end
        if ((minutiae_img(x,y,3) ~= 0) | (minutiae_img(x,y,1) ~= 0))
            combined(x,y,:) = minutiae_img(x,y,:);
        end
    end % end for y
end % end for x

%Calculo do ponto medio das minucias fim de linha
[xmf, ymf] = calculaPontoMedio(linha_fim, minucias_fim);
%Mostra o ponto medio na cor verde na imagem das minucias fim de linha
minutiae_fim(xmf, ymf, :) = [0,255,0];

%Calculo do ponto medio das minucias de bifurcacao
[xmb, ymb] = calculaPontoMedio(linha_bif, minucias_bif);
%Mostra o ponto medio na cor verde na imagem das minucias bifurcadas
minutiae_bif(xmb, ymb, :) = [0,255,0];

%Calcula as distancias euclidiana a partir do ponto de centro de massa das
%minucias fim de linha
% (Fim de linha)
ymax=max(minucias_fim(:,2)); %y maximo da matriz minucias_fim
ymin=min(minucias_fim(:,2)); %y minimo da matriz minucias_fim
xmax=max(minucias_fim(:,1)); %x maximo da matriz minucias_fim
xmin=min(minucias_fim(:,1)); %x minimo da matriz minucias_fim
linha=1;
distancia_fim(linha, :) = [0,0,0,0,0,0];
%1º Quadrante: y variando de ymf ate y minimo e
%          x variando de xmax ate (xmf+1) (considera pontos
%          localizados na reta do x positivo)
for y=ymf:(-1):ymin
    for x=xmax:(-1):(xmf+1)
        if minutiae_fim(x, y, 1) == 255
            d = sqrt((xmf - x)^2 + (ymf - y)^2);
            distancia_fim(linha, :) = [1, xmf, ymf, x, y, d];
            if d > 80
                minutiae_fim(x,y,:) = [250,250,0];
            end
            if d < 10
                minutiae_fim(x,y,:) = [250,0,250];
            end
            linha = linha + 1;
        end %if
    end %y
end %x
%2º Quadrante: x variando de xmf ate x minimo e

```

```

%y variando de y minimo ate ymf (considera pontos localizados
%na reta do y negativo)
for x=xmf:(-1):xmin
    for y=ymin:1:(ymf-1)
        if minutiae_fim(x, y, 1) == 255
            d = sqrt((xmf - x)^2 + (ymf - y)^2);
            distancia_fim(linha, :) = [2, xmf, ymf, x, y, d];
            if d > 80
                minutiae_fim(x,y,:) = [250,250,0];
            end
            if d < 10
                minutiae_fim(x,y,:) = [250,0,250];
            end
            linha = linha + 1;
        end %if
    end %y
end %x
%3º Quadrante: y variando de ymf ate y maximo
%x variando de x minimo ate (xmf-1) (consera pontos localizados na reta do x negativo)
for y=ymf:1:ymin
    for x=xmin:1:(xmf-1)
        if minutiae_fim(x, y, 1) == 255
            d = sqrt((xmf - x)^2 + (ymf - y)^2);
            distancia_fim(linha, :) = [3, xmf, ymf, x, y, d];
            if d > 80
                minutiae_fim(x,y,:) = [250,250,0];
            end
            if d < 10
                minutiae_fim(x,y,:) = [250,0,250];
            end
            linha = linha + 1;
        end %if
    end %y
end %x
%4º Quadrante: x variando de xmf ate x maximo e
%y variando de y maximo ate ymf+1 (considera pontos localizados na reta do y positivo)
for x=xmf:1:xmax
    for y=ymin:(-1):(ymf+1)
        if minutiae_fim(x, y, 1) == 255
            d = sqrt((xmf - x)^2 + (ymf - y)^2);
            distancia_fim(linha, :) = [4, xmf, ymf, x, y, d];
            if d > 80
                minutiae_fim(x,y,:) = [250,250,0];
            end
            if d < 10
                minutiae_fim(x,y,:) = [250,0,250];
            end
            linha = linha + 1;
        end %if
    end %y
end %x

% Ignora os pontos onde a distancia entre a coordenada e o centro de massa
% e superior a 80 e inferior a 10
tamanhoFim=size(distancia_fim,1);
distanciaFim = [0,0,0,0,0,0];
linha=1;
for i=1:tamanhoFim

```



```

if distancia_fim(i,6) >= 10 & distancia_fim(i,6) <= 80
    distanciaFim(linha,:) = [distancia_fim(i,1),distancia_fim(i,2),distancia_fim(i,3),
        distancia_fim(i,4),distancia_fim(i,5),distancia_fim(i,6)];
    linha=linha+1;
end
end
clear i
clear linha
clear tamanhofim
clear distancia_fim

% Calcula as distancias euclidiana a partir do ponto de centro de massa
% (Bifurcacao)
clear ymax
clear ymin
clear xmax
clear xmin
ymax=max(minucias_bif(:,2)); %y maximo da matriz minucias_bif
ymin=min(minucias_bif(:,2)); %y minimo da matriz minucias_bif
xmax=max(minucias_bif(:,1)); %x maximo da matriz minucias_bif
xmin=min(minucias_bif(:,1)); %x minimo da matriz minucias_bif
linha=1;
distancia_bif(linha, :) = [0,0,0,0,0,0];
%1º Quadrante: y variando de ymb ate y minimo e
%x variando de x maximo ate (xmb+1) (considera os pontos
%localizados na reta do x positivo)
for y=ymb:(-1):ymin
    for x=xmax:(-1):(xmb+1)
        if minutiae_bif(x, y, 3) == 255
            d = sqrt((xmb - x)^2 + (ymb - y)^2);
            distancia_bif(linha, :) = [1, xmb, ymb, x, y, d];
            if d > 80
                minutiae_bif(x,y,:) = [0,250,250];
            end
            if d < 10
                minutiae_bif(x,y,:) = [250,0,250];
            end
            linha = linha + 1;
        end %if
    end %y
end %x
%2º Quadrante: x variando de xmb ate x minimo e
%y variando de y minimo ate (ymb-1) (considera os pontos localizados no y negativo)
for x=xmb:(-1):xmin
    for y=ymin:1:(ymb-1)
        if minutiae_bif(x, y, 3) == 255
            d = sqrt((xmb - x)^2 + (ymb - y)^2);
            distancia_bif(linha, :) = [2, xmb, ymb, x, y, d];
            if d > 80
                minutiae_bif(x,y,:) = [0,250,250];
            end
            if d < 10
                minutiae_bif(x,y,:) = [250,0,250];
            end
            linha = linha + 1;
        end %if
    end %y
end %x

```

```

%3º Quadrante: y variando de ymb ate y maximo e
% x variando de xminimo ate (xmb-1) (considera os pontos
% localizados no x negativo)
for y=ymb:1:ymax
    for x=xmin:1:(xmb-1)
        if minutiae_bif(x, y, 3) == 255
            d = sqrt((xmb - x)^2 + (ymb - y)^2);
            if d > 80
                minutiae_bif(x,y,:) = [0,250,250];
            end
            if d < 10
                minutiae_bif(x,y,:) = [250,0,250];
            end
            distancia_bif(linha, :) = [3, xmb, ymb, x, y, d];
            linha = linha + 1;
        end %if
    end %y
end %x
%4º Quadrante: x variando de xmb ate x maximo e
% y variando de y maximo ate (ymb+1) (considera os pontos localizados na reta do y positivo)
for x=xmb:1:xmax
    for y=ymax:(-1):(ymb+1)
        if minutiae_bif(x, y, 3) == 255
            d = sqrt((xmb - x)^2 + (ymb - y)^2);
            distancia_bif(linha, :) = [4, xmb, ymb, x, y, d];
            if d > 80
                minutiae_bif(x,y,:) = [0,250,250];
            end
            if d < 10
                minutiae_bif(x,y,:) = [250,0,250];
            end
            linha = linha + 1;
        end %if
    end %y
end %x

% Ignora os pontos onde a distancia entre a coordenada e o centro de massa
% e superior a 80 e inferior a 10
tamanhobif=size(distancia_bif,1);
distanciaBif = [0,0,0,0,0,0];
linha=1;
for i=1:tamanhobif
    if distancia_bif(i,6) >= 10 & distancia_bif(i,6) <= 80
        distanciaBif(linha,:) = [distancia_bif(i,1),distancia_bif(i,2),distancia_bif(i,3),
                                distancia_bif(i,4),distancia_bif(i,5),distancia_bif(i,6)];
        linha=linha+1;
    end
end
clear i
clear linha
clear tamanhobif
clear distancia_bif

figure
imshow(minutiae_bif), title('minucias de bifurcacao')
figure
imshow(minutiae_fim), title('minucias fim de linha')
figure

```

```

imshow(combined), title('imagem pre-processada e minucias')
figure
imshow(minutiae_img), title('minucias')
figure
imshow(thinned), title('imagem pre-processada')
figure
imshow(img), title('imagem original')

```

2) ordenarVetorRotacionado

```

function [Dixon, espurios, espuriosRotac, erro] = ordenarVetorRotacionado(distancia, distanciaRotac,
ParamDifDistancia)

```

```

%Procura o vetor alinhamento para localizar o ponto inicial para rotacionar
%o vetor que contem as informacoes dos pontos Rotacionados
%Retorna o inicio do vetor rotacionado das minucias
erro = 0;

```

```

[index] = vetorAlinhamento(distancia, distanciaRotac, ParamDifDistancia);

```

```

if index==0
    erro = 1;
    Dixon=[0,0,0,0];
    espurios=[0,0,0,0,0];
    espuriosRotac = [0,0,0,0,0];
else
    %Busca o melhor vetor alinhamento
    [Dixon, espurios, espuriosRotac, erro] = verificaEspurios(distancia, distanciaRotac, index,
ParamDifDistancia);
End

```

2.1) vetorAlinhamento

```

function [index] = vetorAlinhamento(distancia, distanciaRotac, ParamDifDistancia)

```

```

mediaOrigem=0;
passaTesteIndex=0;
Encontrou=0;
vez=1;
vezUtilizada=0;
tamanhoRotac=size(distanciaRotac,1);
while passaTesteIndex == 0
    if vez==1
        mediaOrigem = (distancia(1,6)+distancia(2,6)+distancia(3,6)+distancia(4,6))/4;
    elseif vez==2
        mediaOrigem = (distancia(1,6)+distancia(2,6)+distancia(4,6)+distancia(5,6))/4;
    elseif vez==3
        mediaOrigem = (distancia(1,6)+distancia(3,6)+distancia(4,6)+distancia(5,6))/4;
    elseif vez==4
        mediaOrigem = (distancia(2,6)+distancia(3,6)+distancia(4,6)+distancia(5,6))/4;
    elseif vez==5
        mediaOrigem = (distancia(1,6)+distancia(2,6)+distancia(3,6)+distancia(5,6))/4;
    elseif vez==6
        mediaOrigem = (distancia(3,6)+distancia(4,6)+distancia(5,6)+distancia(6,6))/4;
    end
    [alinhamentoRotac] = vetorAlinhamentoMedia4pontos(distancia, distanciaRotac, mediaOrigem);
    [minval index] = min(alinhamentoRotac(:,2));
    if minval ~= 0
        while minval < 2.5 & passaTesteIndex == 0
            [testeIndex] = verificaIndexEncontrado(distancia, distanciaRotac, ParamDifDistancia, index, vez, 1, 4);
            if size(testeIndex,1) == 4

```

```

        passaTesteIndex = 1;
        Encontrou = 1;
        vezUtilizada = vez;
    else
        alinhamentoRotac(index,2)=9999;
        clear testeIndex;
        testeIndex=[0,0,0];
        [minval index] = min(alinhamentoRotac(:,2));
    end
end
else
    passaTesteIndex=1;
    Encontrou = 1;
    vezUtilizada = vez;
end
vez=vez+1;
%Sai da verificacao se passou todos os tipos de vez
if vez > 6
    passaTesteIndex=1;
end
clear testeIndex;
testeIndex=[0,0,0];
end
%Se nao Encontrou o valor de index eh zero
if Encontrou==1
    %Se encontrou verifica o tipo de vez e se os pontos espurios sao
    %encontrados
    %verifica se o index anterior ao encontrado eh igual ao primeiro
    %index da origem
    if index-1 == 0
        posterior = tamanhoRotac;
    else
        posterior=index-1;
    end
    if index-2 == 0
        antiposterior=tamanhoRotac;
    elseif index-2 < 0
        antiposterior=tamanhoRotac-1;
    else
        antiposterior=index-2;
    end
    if vezUtilizada == 4 & abs(distancia(1,6)-distanciaRotac(posterior,6)) < ParamDifDistancia
        index=posterior;
    elseif vezUtilizada == 6
        if abs(distancia(2,6)-distanciaRotac(posterior,6)) < ParamDifDistancia
            index=posterior;
        end
        if abs(distancia(1,6)-distanciaRotac(antiposterior,6)) < ParamDifDistancia
            index=antiposterior;
        end
    else
        index=index;
    end
end
else
    index = RefazVetorAlinhamento(distancia, distanciaRotac, ParamDifDistancia);
end
end

```

2.1.1) vetorAlinhamentoMedia4pontos

```
function [alinhamentoRotac] = vetorAlinhamentoMedia4pontos(distancia, distanciaRotac, media)
%no vetor alinhamento Rotac sera armazenado o ponto inicial e a
%diferenca entre o valor Media e de todos os pontos da rotacionada de 4 em 4
linha1=0;
linha2=0;
linha3=0;
linha4=0;
linha=1;
alinhamentoRotac = [0,0];
tamanhoRotac = size(distanciaRotac,1);
for i=1:tamanhoRotac
    linha1=i;
    if i==tamanhoRotac-2
        linha2=i+1;
        linha3=i+2;
        linha4=1;
    elseif i==tamanhoRotac-1
        linha2=i+1;
        linha3=1;
        linha4=2;
    elseif i==tamanhoRotac
        linha2=1;
        linha3=2;
        linha4=3;
    else
        linha2=i+1;
        linha3=i+2;
        linha4=i+3;
    end
    mediaRotac = (distanciaRotac(linha1,6)+distanciaRotac(linha2,6)+distanciaRotac(linha3,6)+
        distanciaRotac(linha4,6))/4;
    diferenca = abs(mediaRotac-media);
    alinhamentoRotac(linha,:) = [i, diferenca];
    linha=linha+1;
end
```

2.1.2.) verificaIndexEncontrado

```
function [testaIndex] = verificaIndexEncontrado(distancia, distanciaRotac, ParamDifDistancia, index,
vez, inicio, fim)
```

```
j=index;
testaIndex=[0,0,0];
tamanhoRotac = size(distanciaRotac,1);
linha=1;
jpulado=0;
%Verifica se o index encontrado realmente representa o inicio do vetor
%Alinhamento
for i=inicio:fim
    if vez==1
        diferenca = abs(distancia(i,6)-distanciaRotac(j,6));
        a=i;
    elseif vez==2
        if i>=3
            diferenca = abs(distancia(i+1,6)-distanciaRotac(j,6));
            a=i+1;
        else
```

```

        diferenca = abs(distancia(i,6)-distanciaRotac(j,6));
        a=i;
    end
elseif vez==3
    if i>=2
        diferenca = abs(distancia(i+1,6)-distanciaRotac(j,6));
        a=i+1;
    else
        diferenca = abs(distancia(i,6)-distanciaRotac(j,6));
        a=i;
    end
elseif vez==4
    diferenca = abs(distancia(i+1,6)-distanciaRotac(j,6));
    a=i+1;
elseif vez==5
    if i >=4
        diferenca = abs(distancia(i+1,6)-distanciaRotac(j,6));
        a=i+1;
    else
        diferenca = abs(distancia(i,6)-distanciaRotac(j,6));
        a=i;
    end
elseif vez==6
    diferenca = abs(distancia(i+2,6)-distanciaRotac(j,6));
    a=i+2;
end
b=j;
%ParamDifDistancia eh o parametro para a diferenca das distancias
%utilizadas para fim de linha e bifurcada podem ser valores diferentes
if diferenca < ParamDifDistancia
    testaIndex(linha,:) = [a,b,diferenca];
    linha=linha+1;
    if jpulado == j
        jpulado=0;
        j=j+2;
    else
        j=j+1;
    end
else
    if jpulado==j
        j=j+2;
    else
        if j+1 > tamanhoRotac
            proximo=1;
        else
            proximo=j+1;
        end
        if abs(distancia(i,6)-distanciaRotac(proximo,6)) < ParamDifDistancia
            testaIndex(linha,:) = [a,proximo,abs(distancia(i,6)-distanciaRotac(proximo,6))];
            linha=linha+1;
            jpulado=j;
        else
            j=j+1;
        end
    end
end
end
if j > tamanhoRotac
    j=1;

```

```

end
end

```

2.1.3) refazVetorAlinhamento

```

%Re-procura um vetor de alinhamento das minucias
function [index] = RefazVetorAlinhamento(distancia, distanciaRotac, ParamDifDistancia)
mediaOrigem=0;
passaTesteIndex=0;
Encontrou=0;
vez=1;
vezUtilizada=0;
tamanhoRotac=size(distanciaRotac,1);
testeIndex=[0,0,0];
while passaTesteIndex == 0
    if vez==1
        mediaOrigem = (distancia(5,6)+distancia(6,6)+distancia(7,6)+distancia(8,6))/4;
    elseif vez==2
        mediaOrigem = (distancia(5,6)+distancia(6,6)+distancia(8,6)+distancia(9,6))/4;
    elseif vez==3
        mediaOrigem = (distancia(5,6)+distancia(7,6)+distancia(8,6)+distancia(9,6))/4;
    elseif vez==4
        mediaOrigem = (distancia(6,6)+distancia(7,6)+distancia(8,6)+distancia(9,6))/4;
    elseif vez==5
        mediaOrigem = (distancia(5,6)+distancia(6,6)+distancia(7,6)+distancia(9,6))/4;
    elseif vez==6
        mediaOrigem = (distancia(7,6)+distancia(8,6)+distancia(9,6)+distancia(10,6))/4;
    end
    [alinhamentoRotac] = vetorAlinhamentoMedia4pontos(distancia, distanciaRotac, mediaOrigem);
    [minval index] = min(alinhamentoRotac(:,2));
    if minval ~= 0
        while minval < 2.5 & passaTesteIndex == 0
            [testeIndex] = verificaIndexEncontrado(distancia, distanciaRotac, ParamDifDistancia, index, vez, 5, 8);
            if size(testeIndex,1) == 4
                passaTesteIndex = 1;
                Encontrou = 1;
                vezUtilizada = vez;
            else
                alinhamentoRotac(index,2)=9999;
                clear testeIndex;
                testeIndex=[0,0,0];
                [minval index] = min(alinhamentoRotac(:,2));
            end
        end
    end
    else
        passaTesteIndex=1;
        Encontrou = 1;
        vezUtilizada = vez;
    end
    vez=vez+1;
    %Sai da verificacao se passou todos os tipos de vez
    if vez > 6
        passaTesteIndex=1;
    end
    clear testeIndex;
    testeIndex=[0,0,0];
end
%Se nao Encontrou o valor de index eh zero
if Encontrou==1

```

```

%Se encontrou verifica o tipo de vez e se os pontos espurios sao
%encontrados
%verifica se o index anterior ao encontrado eh igual ao primeiro
%index da origem
if index-1 == 0
    anterior = tamanhoRotac;
else
    anterior=index-1;
end
if index-2 == 0
    antiAnterior=tamanhoRotac;
elseif index-2 < 0
    antiAnterior=tamanhoRotac-1;
else
    antiAnterior=index-2;
end

if vezUtilizada == 4 & abs(distancia(5,6)-distanciaRotac(anterior,6)) < ParamDifDistancia
    index=anterior;
elseif vezUtilizada == 6
    if abs(distancia(6,6)-distanciaRotac(anterior,6)) < ParamDifDistancia
        index=anterior;
    end
    if abs(distancia(5,6)-distanciaRotac(antiAnterior,6)) < ParamDifDistancia
        index=antiAnterior;
    end
end
%Verifica os pontos de 1 a 4
j=index;
linha=1;
jpulado=0;
anterior=j-1;
for i=4:(-1):1
    if anterior <= 0
        j=tamanhoRotac-anterior;
    else
        j=anterior;
    end
    diferenca = abs(distancia(i,6)-distanciaRotac(j,6));
    if diferenca < ParamDifDistancia
        testaIndex(linha,:) = [i,j,diferenca];
        linha=linha+1;
        if jpulado == j
            jpulado=0;
            j=j-2;
        else
            j=j-1;
            anterior=j;
        end
    end
    else
        if jpulado==j
            j=j-2;
            jpulado=0;
        else
            antiAnterior=j-1;
            if antiAnterior <= 0
                antiAnterior=tamanhoRotac-antiAnterior;
            end
        end
    end
end

```



```

        if abs(distancia(i,6)-distanciaRotac(antiAnterior,6)) < ParamDifDistancia
            testaIndex(linha,:) = [i,antiAnterior,abs(distancia(i,6)-distanciaRotac(antiAnterior,6))];
            linha=linha+1;
            jpulado=j;
        else
            j=j-1;
            anterior=j;
        end
    end
end
end
[indexTeste] = min(testaIndex(:,2));
if indexTeste ~= 0
    index=indexTeste;
end
else
    index=0;
end
end

```

2.2) verificaEspurios

```

function [Dixon, espurios, espuriosRotac, erro] = verificaEspurios(distancia, distanciaRotac, index,
                                                                    ParamDifDistancia)

%Ordena o vetor rotacionado iniciando pelo indice encontrado no vetor de
%Alinhamento, armazenando os pontos espurios das minucias da imagem origem
%e da imagem rotacionada em vetores separados
tamanho = size(distancia,1);
tamanhoRotac = size(distanciaRotac,1);
linha=1;
linhae=1;
linhaR=1;
indexPosterior=0;
k=1;
indexPulado=0;
ilgualindexpulado=0;
Dixon=[0,0,0,0];
espurios=[0,0,0,0,0];
espuriosRotac = [0,0,0,0,0];
indexNaoLocalizado=0;
erro=0;
for i=1:tamanho
    if index > tamanhoRotac
        index=1;
    end
    if index+1 > tamanhoRotac
        indexPosterior = 1;
    else
        indexPosterior = index+1;
    end

    %indexPulado eh quando o proximo ja foi utilizado, assim neste caso
    %para as proximas comparacoes adiciona dois no index para nao
    %utiliza-lo novamente

    %indexNaoLocalizado eh quando o proximo NAO foi utilizado, assim neste
    %caso para as proximas comparacoes adiciona um no index para nao
    %ignora-lo
    if indexPulado ~= 0 | indexNaoLocalizado ~= 0
        %Compara o i atual com o index pulado, senao for encontrado entao o

```

```

%indexPulado sera espurio do vetor rotacionado
if (abs(distancia(i,6) - distanciaRotac(index,6)) < ParamDifDistancia)
    Dixon(linha,:)=[index,distanciaRotac(index,1),distanciaRotac(index,6),i];
    linha=linha+1;
    ilgualindexpulado=1;
else
    espuriosRotac(linhaR,:) = [index,distanciaRotac(index,6),distanciaRotac(index,1),i,0];
    linhaR=linhaR+1;
    ilgualindexpulado=0;
end
if indexNaoLocalizado ~= 0
    index=index+1;
else
    index=index+2;
end
indexPosterior=index+1;
%Trata o valor do index para nao estourar a matriz
if index > tamanhoRotac
    index=index-tamanhoRotac;
end
if indexPosterior > tamanhoRotac
    indexPosterior=indexPosterior-tamanhoRotac;
end

indexPulado=0;
indexNaoLocalizado=0;
%Se o i nao foi encontrado como indexPulado, compara-o com o index
%conforme situacao detalhada acima
if ilgualindexpulado ~= 1
    ilgualindexpulado=0;
    if abs(distancia(i,6)-distanciaRotac(index,6)) < ParamDifDistancia
        Dixon(linha,:)=[index,distanciaRotac(index,1),distanciaRotac(index,6),i];
        linha=linha+1;
        index=index+1;
    elseif (abs(distancia(i,6) - distanciaRotac(indexPosterior,6)) < ParamDifDistancia)

Dixon(linha,:)=[indexPosterior,distanciaRotac(indexPosterior,1),distanciaRotac(indexPosterior,6),i];
        linha=linha+1;
        indexPulado = index;
    else
        indexNaoLocalizado = index;
        espurios(linhae,:) = [i,distancia(i,6),distancia(i,1),index,0];
        linhae=linhae+1;
    end
end
% Se o indexNao foi localizado eh o final do for nao conseguira
% perdera o valor desse ultimo espurio
if indexNaoLocalizado ~= 0 & i == tamanho
    espuriosRotac(linhaR,:) = [index,distanciaRotac(index,6),distanciaRotac(index,1),i+1,0];
    linhaR=linhaR+1;
    ilgualindexpulado=0;
end
else
    if (abs(distancia(i,6) - distanciaRotac(index,6)) < ParamDifDistancia)
        Dixon(linha,:)=[index,distanciaRotac(index,1),distanciaRotac(index,6),i];
        linha=linha+1;
        index=index+1;
    elseif (abs(distancia(i,6) - distanciaRotac(indexPosterior,6)) < ParamDifDistancia)

```

```

Dixon(linha,:)= [indexPosterior,distanciaRotac(indexPosterior,1),distanciaRotac(indexPosterior,6),i];
    linha=linha+1;
    indexPulado = index;
else
    indexNaoLocalizado = index;
    espurios(linha,:) = [i,distancia(i,6),distancia(i,1),index,0];
    linha=linha+1;
end
end
% Se o index nao foi localizado eh o final do for nao conseguiu
% perdera o valor desse ultimo espurio
if indexPulado ~= 0 & i == tamanho
    espuriosRotac(linhaR,:) = [index,distanciaRotac(index,6),distanciaRotac(index,1),i+1,0];
    linhaR=linhaR+1;
end
end

diferencaDePontos = (tamanhoRotac - size(espuriosRotac,1)) - (tamanho - size(espurios,1));
if size(espuriosRotac,1) ~= 0 & espuriosRotac(1,1) ~= 0
    ultimoEspurio = espuriosRotac(size(espuriosRotac,1),1);
else
    ultimoEspurio = 0;
end
if size(Dixon,1) ~= 0 & Dixon(1,1) ~= 0
    ultimoDixon = Dixon(size(Dixon,1),1);
else
    ultimoDixon = 0;
end

if ultimoDixon ~= 0
    if ultimoEspurio > ultimoDixon
        index = ultimoEspurio+1;
    elseif ultimoEspurio <= ultimoDixon
        index = ultimoDixon+1;
    end
    if index > tamanhoRotac
        index=1;
    end
    if diferencaDePontos >= 1
        for j=1:diferencaDePontos
            espuriosRotac(linhaR,:) = [index,distanciaRotac(index,6),distanciaRotac(index,1),i,0];
            linhaR=linhaR+1;
            index=index+1;
            if index+1 > tamanhoRotac
                index=1;
            end
        end
    end
else
    erro = 1;
end
end

```

3) reordenarVetorRotacionado

```

function [DixonReordenado, espuriosReordenadoO, espuriosReordenadoR] =
    reordenarVetorRotacionado (Dixon, espuriosOrigem, espuriosRotac, ParamDifDistancia)
tamanhoDixon = size(Dixon, 1);
tamanhoEspuriosOrigem = size(espuriosOrigem, 1);
tamanhoEspuriosRotac = size(espuriosRotac, 1);

```

```

DixonReordenado=[0,0,0,0];
espuriosReordenadoO=[0,0,0,0];
espuriosReordenadoR = [0,0,0,0];
linha=1;
linhaRR=1;
linhaRO=1;
linhaEspuriosOrigem=1;
linhaEspuriosRotac=1;
linhaEncontrada=0;
i=1;
novoSequencial=1;

while i <= tamanhoDixon
    if Dixon(i,4) ~= novoSequencial
        % verifica se a linha nao encontrada em Dixon eh a mesma armazenada
        % em espuriosOrigem
        if linhaEspuriosOrigem > tamanhoEspuriosOrigem
            linhaEspuriosOrigem=tamanhoEspuriosOrigem;
        end
        if espuriosOrigem(linhaEspuriosOrigem, 1) == novoSequencial
            distanciaOrigem=espuriosOrigem(linhaEspuriosOrigem, 2);
            indexOrigem=espuriosOrigem(linhaEspuriosOrigem, 4);
            %tenta localizar a mesma distancia no vetor de espuriosRotac, obedecendo algumas regras
            while linhaEspuriosRotac <= tamanhoEspuriosRotac & linhaEncontrada==0
                %Regras: diferenca entre as distancias inferior a ParamDifDistancia, e a
                %diferença em os index dos pontos sejam inferiores a 5
                if (abs(espuriosRotac(linhaEspuriosRotac, 2) - distanciaOrigem) < ParamDifDistancia) &
                    (abs(espuriosRotac(linhaEspuriosRotac, 1) - indexOrigem) < 5 ) &
                    (espuriosRotac(linhaEspuriosRotac, 5) ~= 1)
                    DixonReordenado(linha,:) = [espuriosRotac(linhaEspuriosRotac,1),
                                                    espuriosRotac(linhaEspuriosRotac, 3),
                                                    espuriosRotac(linhaEspuriosRotac, 2), linhaEspuriosRotac];

                    linha=linha+1;
                    linhaEncontrada=1;
                    espuriosRotac(linhaEspuriosRotac,5) = 1;
                end
                linhaEspuriosRotac=linhaEspuriosRotac+1;
            end % do while
            %armazena os espurios Origem se nao foi encontrado nos espurios
            %rotacionados OU todas as linhas dos espurios rotacionados foram
            %esgotadas
            if linhaEncontrada == 0 | (linhaEspuriosRotac > tamanhoEspuriosRotac)
                espuriosReordenadoO(linhaRO,:) = [espuriosOrigem(linhaEspuriosOrigem,1),
                                                    espuriosOrigem(linhaEspuriosOrigem,2),
                                                    espuriosOrigem(linhaEspuriosOrigem,3),
                                                    espuriosOrigem(linhaEspuriosOrigem,4)];

                linhaRO=linhaRO+1;
            end
            linhaEncontrada=0;
            linhaEspuriosRotac=1;
            linhaEspuriosOrigem=linhaEspuriosOrigem+1;
        end
    else
        DixonReordenado(linha,:) = [Dixon(i,1), Dixon(i,2), Dixon(i,3), Dixon(i,4)];
        linha=linha+1;
        i=i+1;
    end
    novoSequencial=novoSequencial+1;
end

```

```

end

%Reordena os espurios rotacionados, pois no processo apenas foi marcado na
%coluna 5 com valor 1 os espurios que foram re-localizados
linha=1;
i=0;
if espuriosRotac(1,1) ~= 0
    for i=1:size(espuriosRotac,1)
        if espuriosRotac(i,5) == 0
            espuriosReordenadoR(linha,:)= [espuriosRotac(i,1), espuriosRotac(i,2), espuriosRotac(i,3),
                                            espuriosRotac(i,4)];

            linha=linha+1;
        end
    end
end
end

```

4) DixonMedia

function [ordenado] = DixonMedia (Dixon, espurios, distancia)

```

%Substitui o ponto espurio pela media entre os tres pontos existentes vetor
%origem
tamanho = size(Dixon,1);
tamanhoEspurio = size(espurios,1);
%ordenado armazena o quadrante, a media, distancia euclidiana espuria
ordenado = [0,0,0];
linhae=1;
linha=1;
anterior=0;
posterior=0;
for i=1:tamanho
    if i-1 == 0
        anterior = tamanho;
        posterior = i+1;
    elseif i+1 > tamanho
        posterior = 1;
    else
        anterior = i-1;
        posterior = i+1;
    end
    if linhae <= tamanhoEspurio & espurios(linhae,1) == i & espurios(1,1) ~= 0
        %armazena a media no lugar do ponto espurio
        media = (distancia(anterior,6)+distancia(i,6)+distancia(posterior,6))/3;
        ordenado(linha,:) = [espurios(linhae,3), media, espurios(linhae,2)];
        linha=linha+1;
        %armazena o ponto i da DixonFim
        ordenado(linha,:) = [Dixon(i,2), Dixon(i,3), 0];
        linha=linha+1;
        linhae=linhae+1;
    else
        ordenado(linha,:) = [Dixon(i,2), Dixon(i,3), 0];
        linha=linha+1;
    end
end
end

```

5) DistanciaEuclidianaMedia

function [DistanciaEuclidiana] = DistanciaEuclidiana (distancia, ordenado, quantidadeEspurio)

```

tamanhoDistancia = size(distancia,1);

```

```

tamanhoOrdenado = size(ordenado,1);
distanciaDiferenca=0;

%mudança no calculo da Distancia Euclidiana para usar a metodologia "Normal"
% ignora os pontos que forem espurios na diferença da distancia mas no
% final faz o normalizacao utilizando a quantidade de pontos espurios
if tamanhoDistancia <= tamanhoOrdenado
    for i=1:tamanhoDistancia
        distanciaDiferenca = distanciaDiferenca + (distancia(i,6) - ordenado(i,2))^2;
    end
    tamanho = tamanhoDistancia;
else
    for i=1:tamanhoOrdenado
        distanciaDiferenca = distanciaDiferenca + (distancia(i,6) - ordenado(i,2))^2;
    end
    tamanho = tamanhoOrdenado;
end
if quantidadeEspurio >= tamanho
    DistanciaEucliana = 9999999;
else
    diferenca = tamanho - quantidadeEspurio;
    divisao = tamanho/diferenca;
    DistanciaEucliana = (divisao) * (sqrt(distanciaDiferenca));
end

```

6) DistanciaEuclidianaNormal

function [DistanciaEucliana] = DistanciaEuclidiana (distancia, ordenado, quantidadeEspurio)

```

tamanhoDistancia = size(distancia,1);
tamanhoOrdenado = size(ordenado,1);
distanciaDiferenca=0;

%mudança no calculo da Distancia Euclidiana para usar a metodologia "Normal"
% ignora os pontos que forem espurios na diferença da distancia mas no
% final faz o normalizacao utilizando a quantidade de pontos espurios
if tamanhoDistancia <= tamanhoOrdenado
    for i=1:tamanhoDistancia
        if ordenado(i,3) == 0
            distanciaDiferenca = distanciaDiferenca + (distancia(i,6) - ordenado(i,2))^2;
        end
    end
    tamanho = tamanhoDistancia;
else
    for i=1:tamanhoOrdenado
        if ordenado(i,3) == 0
            distanciaDiferenca = distanciaDiferenca + (distancia(i,6) - ordenado(i,2))^2;
        end
    end
    tamanho = tamanhoOrdenado;
end
if quantidadeEspurio >= tamanho
    DistanciaEucliana = 9999999;
else
    diferenca = tamanho - quantidadeEspurio;
    divisao = tamanho/diferenca;
    DistanciaEucliana = (divisao) * (sqrt(distanciaDiferenca));
end

```