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

EVANDRO DE ARAÚJO JARDINI

**MFIS: Algoritmo de Reconhecimento e Indexação em Base de Dados de
Impressões Digitais em Espaço Métrico**

São Carlos

2007

EVANDRO DE ARAÚJO JARDINI

**MFIS: Algoritmo de Reconhecimento e Indexação em Base de Dados de
Impressões Digitais em Espaço Métrico**

Tese apresentada à Escola de Engenharia de São Carlos, da Universidade de São Paulo, como parte dos requisitos para obtenção do título de Doutor em Engenharia Elétrica

Área de Concentração: Processamento de Sinais de Instrumentação.

Orientador: Prof. Dr. Adilson Gonzaga

São Carlos

2007

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

Dedicatória

A minha mãe Yone, a minha irmã Vanessa e ao meu pai Dermival (em memória)

AGRADECIMENTOS

Em primeiro lugar a Deus, pela ajuda em tudo que consegui e pelo apoio espiritual, confortando minha alma nos momentos difíceis.

Ao meu orientador prof. Dr. Adilson Gonzaga, por ter acreditado neste trabalho e que, sem seu apoio, este não se realizaria.

Ao prof. Dr. Caetano Traina Jr., que através de seu apoio, sugestões e conselhos, possibilitou o desenvolvimento deste trabalho.

Aos companheiros Humberto e ao Enzo pela ajuda no entendimento da *Slim-tree*.

A Gustavo de Sá pela força no entendimento da Curva ROC.

Aos amigos de república Basílio, Paulão, Luiz e Luiz Gustavo (eterno bicho).

A Aline Rezende por sua compreensão.

E a todos que direta e indiretamente contribuíram para o término do trabalho.

RESUMO

JARDINI, E. A. **MFIS: Algoritmo de Reconhecimento e Indexação em Base de Dados de Impressões Digitais em Espaço Métrico**. 2007. 105 f. Tese (Doutorado) - Escola de Engenharia de São Carlos, Departamento de Engenharia Elétrica, Universidade de São Paulo, 2007.

O problema dos métodos tradicionais de identificação de pessoas é que são baseados em senhas e assim podem ser esquecidas, roubadas, perdidas, copiadas, armazenadas de maneira insegura e até utilizadas por uma pessoa que não tenha autorização. Os sistemas biométricos automáticos surgiram para oferecer uma alternativa para o reconhecimento de pessoas com maior segurança e eficiência. Uma das técnicas biométricas mais utilizadas é o reconhecimento de impressões digitais. Com o aumento do uso de impressões digitais nestes sistemas, houve o surgimento de grandes bancos de dados de impressões digitais, tornando-se um desafio encontrar a melhor e mais rápida maneira de recuperar informações.

De acordo com os desafios apresentados, este trabalho tem duas propostas: i) desenvolver um novo algoritmo métrico para identificação de impressões digitais e ii) usá-lo para indexar um banco de dados de impressões digitais através de uma árvore de busca métrica.

Para comprovar a eficiência do algoritmo desenvolvido foram realizados testes sobre duas bases de imagens de impressões digitais, disponibilizadas no evento *Fingerprint Verification Competition* dos anos de 2000 e 2002. Os resultados obtidos foram comparados com os resultados do algoritmo proposto por Bozorth. A avaliação dos resultados foi feita pela curva *Receiver Operating Characteristic* juntamente com a taxa de *Equal Error Rate*, sendo que, o método proposto, obteve a taxa de 4,9% contra 7,2% do método de Bozorth e de 2,0% contra 2,7% do Bozorth nos bancos de dados dos anos de 2000 e 2002 respectivamente. Nos testes de robustez, o algoritmo proposto conseguiu identificar uma impressão digital com uma parte da imagem de apenas 30% do tamanho original e por se utilizar uma base de dados indexada, o mesmo obteve vantagens de tempo na recuperação de pequenas quantidades de impressões digitais de uma mesma classe.

Palavra-Chave: Biometria, reconhecimento de impressões digitais, indexação de impressões digitais e espaço métrico

ABSTRACT

JARDINI, E. A. **MFIS: Algorithm for the Recognition and Indexing in Database of Fingerprints in Metric Spaces**. 2007. 105 f. Thesis (PhD) - Escola de Engenharia de São Carlos, Departamento de Engenharia Elétrica, Universidade de São Paulo, 2007.

The problem of the traditional methods of people identification is that they are based on passwords which may to be forgotten, stolen, lost, copied, stored in an insecure way and be used by unauthorized person. Automatic biometric systems appeared to provide an alternative for the recognition of people in a more safe and efficiency way. One most biometrics techniques used is the fingerprint recognition. With the increasing use of fingerprints in biometric systems, large fingerprint databases emerged, and with them, the challenge to find the best and fastest way to recover informations.

According to the challenges previously mentioned, this work presents two proposals: i) to develop a new metric algorithm for the identification of fingerprints and ii) to use it to index a fingerprint database using a metric search tree.

To prove the efficiency of the developed algorithm tests were performed on two fingerprint images databases from Fingerprint Verification Competition of years 2000 and 2002. The obtained results were compared to the results of the algorithm proposed by Bozorth and was evaluated by the Receiver Operating Characteristic curve and the Equal Error Rate, where the proposed method is of 4.9% against 7.2% of Bozorth and 2.0% of the algorithm proposed against 2.7% of the Bozorth in the databases of the years of 2000 and 2002. In the robustness tests, the proposed algorithm as able to identify a fingerprint with only 30% of the original size and when using an a indexed database, it obtained better performance in the recovery of small amounts of fingerprints of a single class.

Keywords: Biometrics, fingerprint recognition, fingerprint indexing and metric space.

Lista de Figuras

2.1	Participação de cada técnica biométrica em negócios no ano de 2003 (VAUGHAN-NICHOLS, 2004)	7
2.2	Íris	9
3.1	Exemplo de minúcias: Crista Final (marcada com círculo), Crista Bifurcada (marcada com quadrado)	13
3.2	Impressões digitais: a) coletada em ambiente controlado b) latente	15
3.3	Minúcias detectadas pelo software <i>mindtct</i>	16
3.4	Fases do processo de detecção de minúcias.	17
3.5	Mapa direcional de uma impressão digital	18
3.6	Resultado da binarização	19
3.7	Padrões usados para detecção de minúcias	20
3.8	Características de uma minúcia extraídas por detectores de minúcias	21
3.9	<i>Triplet of minutiae</i> : (a) Um <i>triplet</i> sobre uma imagem de impressão digital afi- nada. (b) Um <i>triplet</i> formando um triângulo sem a imagem da impressão digital	26
4.1	Exemplo de consultas de nível 1 para a imagem da Lena (ZACHARY; IYENGAR; BARTHEN, 2001)	30

4.2	Exemplo de consultas de nível 2 para a imagem da Lena (ZACHARY; IYENGAR; BARTHEN, 2001)	30
4.3	Exemplo de consultas de nível 3 para a imagem da Lena (ZACHARY; IYENGAR; BARTHEN, 2001).	30
5.1	Exemplo de <i>Consulta por Abrangência</i> : A figura ilustra uma consulta por abrangência - $RQ(o_q, r_q)$ em um espaço métrico bidimensional utilizando uma função Euclidiana E_1 . O objeto o_q é o objeto de busca enquanto os objetos cinza constituem os objetos do conjunto resposta A.	38
5.2	Exemplo de Consulta aos k -Vizinhos Mais Próximos - $kNNQ(o_q, 4)$ em um espaço bidimensional utilizando a função de distância euclidiana E_1 como a função de distância. O objeto o_q é o objeto de busca enquanto os objetos cinzas constituem os objetos do conjunto resposta A.	39
5.3	Funcionamento do algoritmo MST: a) Nó antes de divisão; b) MST construída sobre os objetos do nó; c) Nó depois da divisão	45
6.1	Diagrama da Metodologia Proposta	52
6.2	Representação da busca de impressões digitais em espaço métrico.	53
6.3	Formato de um arquivo XML representando uma impressão digital	54
6.4	Detalhamento do vetor de características	56
6.5	Vizinhaça de uma minúcia.	57
6.6	Impressão digital com deslocamento de posição	58
6.7	Algoritmo da Função Computacional de Distância Métrica	59
6.8	Ângulos calculados entre as minúcia alvo e minúcias vizinhas.	62
6.9	Conjunto de minúcias em forma de triângulo usado para o cálculo dos ângulos.	62

6.10	Minúcias nas posições originais.	63
6.11	Novas coordenadas das minúcias e seus respectivos ângulos	63
6.12	Exemplo das Imagens do Banco de Dados de 2000 do ano FVC	64
6.13	Exemplo das Imagens do Banco de Dados do ano 2002 do FVC	65
6.14	Curva ROC em escala loglog comparando os resultados entre os sistemas MFIS e o Bozorth3 sobre o banco de dados DB1 FVC 2000	66
6.15	Curvas ROC em escala a) normal e b) ampliada comparando os resultados entre os sistemas MFIS e Bozorth3 sobre o banco de dados DB1 FVC do ano 2000 .	67
6.16	Curvas ROC em escala loglog comparando os resultados entre os sistemas MFIS e Bozorth3 sobre o banco de dados DB1 FVC do ano 2002	68
6.17	Curvas ROC em escala a) normal e b) ampliada comparando os resultados entre os sistemas MFIS e Bozorth3 sobre o banco de dados DB1 FVC 2002	69
6.18	Exemplo de imagens da classe 10 do FVC2002 divididas nas porcentagens de a) 20%, b) 30%, c) 40%, d)50%, e) 60%, f) 70%, g) 80% e h) 90% da imagem original	71
6.19	Porcentagem de acerto do MFIS dos banco de dados FVC2000 e FVC2002. . .	72
6.20	Resultados dos testes de desempenho das classes 10 e 30 da base de dados DB1 do FVC2000	74
6.21	Resultados dos testes de desempenho das classes 50, 70 e 90 da base de dados DB1 do FVC2000	75
6.22	Resultados dos testes de desempenho das classes 10 e 30 da base de dados DB1 do FVC2002	76
6.23	Resultados dos testes de desempenho das classes 50, 70 e 90 da base de dados DB1 do FVC2002	76

6.24 Média dos testes de desempenho de todas as classes das bases de dados DB1 do FVC2000 e FVC2002	77
--	----

Lista de Tabelas

2.1	Aplicações de Reconhecimento de Faces	9
2.2	Comparação entre características biométricas (LIU; SILVERMAN, 2001)	10
5.1	Conjunto de dados usado nos testes de desempenho da <i>Slim-Tree</i> (TRAINA Jr., C.; TRAINA, A.; SEEGER, B.; FALOUTSOS C., 2000).	47
6.1	Taxa EER entre os softwares MFIS e o Bozorth	73

Lista de Abreviaturas

BDI: Banco de Dados de Imagens.

CBIR: *Content-Based Image Retrieval* - Recuperação de Imagem Baseado em Conteúdo.

EER: *Equal Error Rate*

FAR: *False Accept Rate*

FRR: *False Reject Rate*

HSV: *Hue, Saturation, Value* - Matiz, Saturação e Valor. Padrão de espaço de cor criado por A. R. Smith em 1978. O padrão também é conhecido como HSI.

MA: Método de Acesso.

MAE: Método de Acesso Espacial.

MAM: Método de Acesso Métrico.

MFIS: *Metric Fingerprint Identification System* ou Sistema de Identificação de Impressões Digitais Métrico

NFIS: *NIST Fingerprint Image Software*.

NIST: *National Institute of Standards and Technology*.

PWH: *Perceptually Weighted Histogram*.

QBE: *Query by Example* - Consulta por Exemplo.

ROC: *Receiver Operating Characteristic*

XML: *Extensible Markup Language*.

Sumário

1	Introdução	1
1.1	Motivações	3
1.2	Proposta do Trabalho	4
2	Biometria	5
2.1	Introdução	5
2.2	Técnicas Biométricas	7
2.2.1	Impressão Digital	7
2.2.2	Íris	8
2.2.3	Face	8
2.3	Comparação entre os Tipos de Técnicas Biométricas	9
2.4	Estatísticas de Desempenho	10
2.5	Considerações Finais	11
3	Impressão Digital	12
3.1	Introdução	12
3.2	O software <i>Mindtct</i>	15

3.2.1	Mapa Direcional	16
3.2.2	Binarização da Imagem	18
3.2.3	Detecção de Minúcias	20
3.2.4	Contagem de Minúcias Vizinhas	21
3.3	Pesquisas com Impressões Digitais	21
3.3.1	Algoritmos de Identificação de Impressões Digitais	22
3.3.2	Indexação de Impressões Digitais	23
3.4	Considerações Finais	25
4	Recuperação de Imagens Baseada em Conteúdo	27
4.1	Introdução	27
4.2	Contextualização de CBIR	28
4.2.1	Níveis de consultas	29
4.2.2	Pesquisas em CBIR	31
4.3	Índices para Recuperação de Imagens	32
4.4	Considerações Finais	33
5	Indexação em Banco de Dados de Imagens	34
5.1	Introdução	34
5.2	Métodos de Acesso Espaciais	35
5.3	Métodos de Acesso Métricos	36
5.4	Consultas por similaridade	37
5.5	Indexação de Dados em Domínio Métrico	39

5.5.1	<i>A Slim-Tree</i>	41
5.5.1.1	Construindo uma <i>Slim-Tree</i>	43
5.5.1.2	Consultas por Similaridade na <i>Slim-Tree</i>	44
5.5.1.3	Otimização de Sobreposição	45
5.5.1.4	Desempenho da <i>Slim-Tree</i>	46
5.6	Considerações Finais	47
6	Metodologia e Resultados	49
6.1	Introdução	49
6.2	Metodologia	49
6.2.1	Vetor de Características	51
6.2.1.1	Definição das Características	51
6.2.1.2	Criação e Armazenamento do Vetor de Características	55
6.2.2	Algoritmo Proposto	56
6.2.2.1	Triangulação	60
6.3	Material	61
6.4	Resultados	63
6.4.1	Curvas ROC (Receiver Operating Characteristic)	64
6.4.2	Robustez de Recuperação das Imagens	70
6.4.3	Teste de desempenho	73
6.5	Considerações Finais	78

<i>SUMÁRIO</i>	x
7 Conclusões e Trabalhos Futuros	79
7.1 Contribuições da Tese	80
7.2 Sugestões de Trabalhos Futuros	81
Referências Bibliográficas	82

Capítulo 1

Introdução

A era da informação está revolucionando rapidamente o modo como as transações são realizadas. Diariamente, mais e mais atividades comerciais estão sendo realizadas eletronicamente, dispensando o uso do papel e caneta. Este crescimento rápido do comércio eletrônico tem obrigado o uso de apurados sistemas de identificação e autenticação.

Tradicionalmente, os dois tipos de métodos para identificação pessoal automática são (BRAGHIN, 2001) baseados em conhecimento e baseados em objeto. Métodos baseados em conhecimentos usam "alguma coisa que o usuário conhece" para identificá-lo, como uma senha. Métodos baseados em objetos usam "alguma coisa que o usuário possui", como *smart cards*, cartões magnéticos e chaves físicas. A fraqueza desses sistemas é o fato que senhas podem ser esquecidas, compartilhadas ou observadas. Já os objetos podem ser perdidos, roubados, duplicados ou esquecidos em casa. Em adição, eles não são capazes de diferenciar entre uma pessoa autorizada e um impostor.

Tecnologias biométricas são métodos automáticos de reconhecimento de pessoas com base em suas características fisiológicas ou comportamentais. Exemplos dessas características físicas são impressão digital, face, retina, íris, etc. O uso de biometria para identificação e autenticação de pessoas garante muito mais segurança, pois o próprio indivíduo é sua senha.

No passado, as técnicas biométricas eram empregadas para a identificação criminal e segurança nas prisões, mas a partir do momento em que a tecnologia se tornou barata e altamente precisa, sua adoção está sendo feita por uma grande faixa de aplicações comerciais, como o comércio eletrônico e área de controle de acesso.

A autenticação biométrica é realizada por um sistema biométrico, ou seja, um dispositivo automático para verificar ou reconhecer a identidade de uma pessoa com base em suas características fisiológicas.

Os sistemas biométricos comparam as características biométricas do indivíduo com as características armazenadas em um banco de dados.

As características biométricas podem ser obtidas a partir de:

- **Impressões digitais:** São únicas, sendo distintas até para gêmeos idênticos, e não mudam com o decorrer da vida do indivíduo. Seu reconhecimento é feito através de comparações com pontos característicos denominados de minúcias. Possuem desvantagem como cortes e cicatrizes. É a técnica biométrica mais utilizada.
- **Face:** Sistemas de reconhecimento facial são baseados na distância entre os atributos da face (como os olhos) ou sobre a dimensão destes atributos (como o comprimento da boca). Sua fraqueza é que é sensível a variações de iluminação, pose, expressão, etc. Gêmeos idênticos são difíceis de serem distinguidos.
- **Geometria da mão:** A geometria da mão mede a formato da mão. O sistema compara o topo e os lados da mão usando uma câmera. Uma das desvantagens desta técnica é que o formato da mão varia no decorrer da vida do indivíduo.
- **Íris:** Varredura da íris é a mais promissora das técnicas biométricas (NEGIN; CAMUS, 2000). Esta técnica é baseada na varredura do anel colorido em torno da pupila do olho humano. Depois do DNA, as íris são as características mais individualizadas do corpo

humano. Usa-se uma câmera de vídeo durante o processo de varredura e não requer contato entre o olho e o dispositivo biométrico.

Dependendo da técnica biométrica utilizada para o reconhecimento do indivíduo, uma grande quantidade de imagens é gerada e deveser armazenada para posteriormente ser recuperada.

1.1 Motivações

Operações envolvendo a recuperação de imagens possuem dois grandes desafios que são i) recuperar a imagem correta e ii) rapidez no processo de recuperação.

Estes desafios pertencem ao campo de pesquisa conhecido como *Content-Based Image Retrieval* (CBIR) ou Recuperação de Imagens Baseada em Conteúdo. O motivo deste campo de pesquisa é desenvolver técnicas automáticas que possibilitem a recuperação das imagens baseadas somente em seu próprio conteúdo, apesar de existirem consultas que necessitem da interação humana.

Para obter a rapidez no processo de busca das imagens armazenadas, são utilizadas estruturas especiais denominadas de índices. Após se extraírem as características, estas devem ser organizadas de tal forma a permitir sua localização o mais rápido possível. Quanto maior for o banco de dados, maior é a necessidade de velocidade na busca das imagens. Índices para recuperação de imagens recaem sobre dois métodos. Um é denominado Método de Acesso Espacial (MAE), no qual os objetos armazenados são localizados por suas coordenadas. O outro método é o Método de Acesso Métrico (MAM), cuja localização dos objetos é baseada na distância métrica existentes entre os objetos.

1.2 Proposta do Trabalho

De acordo com estes dois desafios citados, este trabalho apresenta propostas de soluções para reconhecimento e busca envolvendo a técnica biométrica de impressão digital. Os objetivos são desenvolver um algoritmo para o reconhecimento de impressões digitais em espaço métrico e adaptá-lo à uma estrutura de indexação métrica.

Esta tese está dividida nos seguintes capítulos:

No capítulo dois são apresentados conceitos referentes a biometria. São descritas pesquisas envolvendo duas técnicas biométricas muito utilizadas: íris e a face. Também é feito um comparativo rápido entre as diversas técnicas biométricas disponíveis.

No capítulo três é dado destaque à técnica biométrica de impressão digital. Discutem-se as pesquisas envolvendo esta técnica, além de se apresentar uma área de pesquisa recente que é a indexação de impressões digitais em banco de dados.

No capítulo quatro são explanados os conceitos de CBIR. São descritos os níveis de consultas existentes, além das pesquisas existentes nesta área.

O capítulo cinco é dedicado às estruturas de indexação, tanto as MAEs quanto as MAMs, sendo que estas últimas são mais detalhadas.

No capítulo seis, é explicada a metodologia proposta juntamente com os resultados obtidos e as discussões sobre eles.

Finalmente, no capítulo sete, é feita a conclusão do trabalho apresentando suas contribuições e sugerindo trabalhos futuros.

Capítulo 2

Biometria

2.1 Introdução

Os métodos tradicionais de reconhecimento e validação de indivíduos recaem sobre um identificador e uma senha. Estes métodos, apesar de muito utilizados, possuem graves problemas que sugerem sua troca por outros. Alguns problemas que podem ser citados são o esquecimento, roubo, cópia indevida, perda, etc. Muitos usuários de computador, por exemplo, menosprezam uma situação de ataque via rede e não se preocupam em tomar atitudes básicas de segurança, como senhas seguras, configuração correta de arquivos compartilhados via rede, etc. O resultado disso são arquivos danificados ou roubados, débito indevido na conta bancária, etc. O ideal é o uso de uma tecnologia que torne difícil uma pessoa não autorizada se passar por outra. Devido a estes problemas, as características biométricas vêm sendo empregadas em diversas situações através de equipamentos, algoritmos, técnicas de extração de características, etc.

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

Segundo Jain et al (JAIN et al., 1997), teoricamente, qualquer característica fisiológica ou comportamental pode ser usada para fazer a identificação de um indivíduo, desde que satisfaça aos seguintes requisitos: i) *universalidade*: significa que todo ser humano deve possuir a característica; ii) *unicidade*, indica que duas pessoas não devem ser as mesmas em termos desta característica; iii) *permanência*, significa que a característica deve ser invariante no tempo e iv) *coletabilidade*¹, que indica que a característica pode ser medida de modo quantitativo.

Tendo em vista a importância de se usar características biométricas presentes no indivíduo como mecanismo de identificação, criou-se uma série de técnicas para extração destas características. Estas técnicas são amplamente usadas nos mais diversos segmentos da sociedade para garantir segurança e precisão no que se refere à capacidade de distinguir entre um indivíduo autorizado e um impostor. Dentre as aplicações das tecnologias biométricas tem-se (RATHA; CHEN; JAIN, 1995): controle de acesso para instalações de alta segurança, verificação do uso de cartões de créditos, identificação de funcionários, etc

Existem diversas técnicas de biometria disponíveis atualmente. Algumas delas ainda estão em estudos, dentro dos laboratórios. Outras já se encontram disponíveis para uso comercial. É difícil determinar o número preciso das técnicas biométricas disponíveis, pois as mesmas evoluem constantemente. Pode-se dizer que as principais são (JAIN et al., 1997; MATYAS Jr.; RIHA, 2000; ROSS; JAIN; QIAN, 2003): impressão digital, geometria de mão, íris, padrões de retina, reconhecimento facial, comparação da voz e assinatura. A figura 2.1 representa a participação que cada técnica teve sobre o montante de US\$ 928 milhões gerados em negócios no ano de 2003 envolvendo biometria (VAUGHAN-NICHOLS, 2004) .

Os sistemas biométricos são utilizados com duas finalidades: de identificação e de verificação (PHILLIPS et al., 2000).

Em sistemas de identificação, uma assinatura biométrica de uma pessoa desconhecida é apresentada ao sistema. O sistema compara a nova assinatura biométrica com um banco de dados

¹Termo adotado para a característica *collectability*

Figura 2.1: Participação de cada técnica biométrica em negócios no ano de 2003 (VAUGHAN-NICHOLS, 2004)

de assinaturas biométricas de indivíduos conhecidos. O sistema reporta ou estima a identidade da pessoa desconhecida deste banco de dados. Sistemas de identificação incluem aqueles que a polícia usa para identificar pessoas baseados em impressão digital.

Em sistemas de verificação, um usuário apresenta uma assinatura biométrica e alega que uma identidade particular pertence àquela assinatura. O algoritmo de verificação irá aceitar ou rejeitar a alegação. Sistemas de verificação incluem aqueles que autenticam a identidade durante uma transação em um ponto de venda ou aqueles que controlam o acesso a determinado local.

2.2 Técnicas Biométricas

2.2.1 Impressão Digital

Impressão digital é o método biométrico mais popular que existe (RATHA; CHEN; JAIN, 1995; VAUGHAN-NICHOLS, 2004). O uso da impressão digital como método biométrico é datado

desde a china antiga para identificar positivamente o autor de um documento (MATYAS Jr.; RIHA, 2000). Em 1960, um estudo feito pelos laboratórios Sandia nos Estados Unidos comparou várias tecnologias biométricas e constatou que a tecnologia de impressão digital tinha o maior potencial para produzir o mais apurado mecanismo de identificação (MATYAS Jr.; RIHA, 2000). Hoje o estudo necessita de atualização, mas serviu para que a tecnologia ganhasse foco nas pesquisas.

Uma vez que a tecnologia de impressão digital é o ponto central deste trabalho, ela será detalhada no próximo capítulo.

2.2.2 Íris

O estudo da íris como método biométrico teve início em 1987 com os oftalmologistas *Flom* e *Aran Safir* que juntamente com o cientista da computação *John Daugman* da Universidade de Cambridge da Inglaterra desenvolveram um software para o reconhecimento de íris (TISSE et al., 2002).

A íris é o anel colorido em torno da pupila dos olhos (figura 2.2). Ela começa a ser formada a partir do primeiro ano de vida da criança e permanece imutável durante a vida do indivíduo. Assim como as impressões digitais, a íris possui características únicas para cada indivíduo. Seus padrões diferem até de olho para olho do mesmo indivíduo.

2.2.3 Face

Os métodos para o reconhecimento de faces humanas têm papel importante em sistemas biométricos e existem diversas pesquisas sobre o tema. Segundo Zhao (ZHAO et al., 2000) existem dois motivos para isso: *i)* a vasta gama de possibilidades comerciais que podem utilizar a tecnologia e *ii)* a grande quantidade de aplicações comerciais existentes devido a mais de trinta

Figura 2.2: Íris

anos de pesquisa. A tabela 2.1 lista algumas das aplicações que fazem uso do reconhecimento de faces.

Tabela 2.1: Aplicações de Reconhecimento de Faces

Áreas	Aplicações
Entretenimento	Vídeo game, realidade virtual, programas de treinamentos, interação homem-máquina
Cartões inteligentes	Carteira de habilitação, passaporte, título de eleitor
Segurança	Dispositivos de <i>logon</i> , segurança em banco de dados, registros médicos, controle de acesso à Internet

Em adição a estas razões tem-se o fato da identificação de faces ser um método denominado não intrusivo, ou seja, assim como no caso da íris, o reconhecimento da face necessita que o usuário posicione em frente de uma câmera, sem ter contato físico com o equipamento.

2.3 Comparação entre os Tipos de Técnicas Biométricas

Como a biometria é uma área de interesse de pesquisa tanto acadêmica quanto comercial, a cada dia surgem diversas novas metodologias. Para se ter uma noção geral das diferenças entre

as técnicas disponíveis são apresentadas na tabela 2.2 as principais técnicas pesquisadas atualmente. Nesta comparação foram avaliadas 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 estes parâmetros de comparação, a impressão digital apresenta-se como uma das características mais adequadas como técnica de sistemas de reconhecimento.

Tabela 2.2: Comparação entre características biométricas (LIU; SILVERMAN, 2001)

Características	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 e idade	Acidentes e idade	Óculos	Pouca luz	Iluminação, idade, cabelo, etc	Mudança na assinatura	Ruído, resfriado, etc
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

2.4 Estatísticas de Desempenho

Independente da técnica biométrica escolhida, o desempenho nos sistemas biométricos deve ser levado em consideração de acordo com sua finalidade. A principal medida de desempenho para sistemas de identificação é a habilidade de identificar o proprietário de uma assinatura biométrica. Mais especificamente, a medida do desempenho é igual à porcentagem de consultas em que a resposta correta possa ser encontrada no pequeno conjunto retornado do banco de dados como resposta.

O desempenho de um sistema de verificação é caracterizado tradicionalmente por duas estatísticas de erro: *taxa de falso-negativo* e *taxa de falso-positivo*. Estas duas taxas são avaliadas em conjunto. Um falso-negativo ocorre quando um sistema rejeita uma identidade válida; um falso-positivo ocorre quando o sistema incorretamente aceita uma identidade.

Em um sistema biométrico perfeito, ambas as taxas deveriam ser zero. Entretanto, como sistemas biométricos não são perfeitos, para cada aplicação deve-se determinar de quais finalidades o usuário está necessitando. Se a necessidade for restringir o acesso para todos, a taxa de falso-negativo deverá ser 100% e falso-positivo deverá ser zero. Agora, se houver a necessidade de permitir o acesso a todos, a taxa de falso-negativo deverá ser zero e a taxa falso-positivo deverá ser 100%.

2.5 Considerações Finais

Áreas como propriedade de documentos, controle de acesso a edifícios ou a construções, autorização de transações comerciais, etc levantaram questões de como garantir a identidade dos indivíduos de um modo confiável e robusto. A biometria se apresenta como modo de prover técnicas para alcançar esses objetivos e, sendo assim, incentivando ainda mais as pesquisas de novas técnicas e metodologias.

Capítulo 3

Impressão Digital

3.1 Introdução

A impressão digital é formada por sulcos presentes nos dedos. A parte alta dos sulcos é denominada crista e a baixa denominada vale. Para o reconhecimento de impressões digitais, as cristas apresentam as características desejadas. Seguindo o fluxo das cristas, nota-se a formação dos pontos característicos usado para a identificação de indivíduos, as minúcias. Dos tipos existentes de minúcias, os dois mais utilizados para o reconhecimento de impressões digitais são: i) minúcia do tipo cristas finais e ii) minúcia do tipo cristas bifurcadas (figura 3.1).

A impressão digital é a técnica biométrica mais usada e antiga (VAUGHAN-NICHOLS, 2004) que se conhece. Os fatores que contribuem para sua larga adoção são i) a formação, que normalmente ocorre no primeiros meses de vida da criança; ii) a imutabilidade garantida durante a vida do indivíduo; iii) a formação ímpar de cada impressão digital, garantindo a unicidade dos indivíduos; iv) ser um método cuja coleta e verificação não é intrusivo para o indivíduo; v) devido a presença de gorduras existentes nas mãos do indivíduo, principalmente nas pontas dos dedos, a impressão digital pode facilmente ser marcada em um objeto segurado por uma pessoa e que pode ser usado para identificá-la; vi) a tecnologia de impressão digital é considerada mais

precisa que outras técnicas biométricas como assinatura e faces, por exemplo (JAIN; HONG, 1996).

Figura 3.1: Exemplo de minúcias: Crista Final (marcada com círculo), Crista Bifurcada (marcada com quadrado)

A literatura de trabalhos biométricos realizados sobre impressões digitais é vasta. Frequentemente, as pesquisas recaem sobre as áreas de classificação de impressões digitais (MAIO; MALTONI, 1996; BALLAN; SAKARYA, 1998; JAIN; PANKANTI, 2000), detecção de minúcias (ZHAO; TANG, 2002; HONG; WAN; JAIN, 1998) e processo de reconhecimento automático de impressões digitais (JAIN et al., 1997, 2000). Outra área que vem surgindo e é um dos temas principais deste trabalho é o processo de indexação de bancos de dados formados por impressões digitais (GERMAIN; CALIFANO; COLVILLE, 1997; BHANU; TAN, 2003), visto que existem grandes quantidades de impressões digitais e existem a necessidade de localização rápida. Maiores detalhes sobre as áreas de reconhecimento e indexação serão apresentados na seção 3.3.

Das três áreas tradicionais de pesquisa, o interesse deste trabalho recai sobre o reconhecimento automático de impressões digitais. Este processo, assim como as de outras técnicas biométri-

cas, concentra-se em duas classes de problemas (JAIN et al., 1997; MATYAS Jr.; RIHA, 2000). A primeira classe, conhecida como verificação, envolve a situação a qual é preciso verificar a autenticidade da identidade do indivíduo. O usuário informa quem ele é e o sistema verifica sua identidade, negando ou aceitando o indivíduo. Para isso, o mesmo deve estar cadastrado em um banco de dados ou as informações pertinentes a ele pode estar armazenadas, por exemplo, em seu cartão digital. Outra classe, a *identificação*, envolve um trabalho mais difícil. A dificuldade desta classe é maior, comparada com a verificação, pois o indivíduo não informa quem ele é, e sim o sistema, através das características biométricas presentes na impressão digital do indivíduo, deve percorrer um banco de dados contendo diversos *templates*¹ de impressões digitais e informar quais impressões mais se assemelham com a impressão de entrada. Uma das dificuldades desse processo dá-se pela falta de qualidade encontrada nas imagens de impressões digitais. A dificuldade aumenta quando a aquisição é realizada em ambientes sem controle, como a cena de um crime.

Em uma cena de crime, as impressões digitais são deixadas por descuido do indivíduo. Frequentemente, são invisíveis para o olho humano sem algum tipo de processo químico. Estas impressões digitais coletadas em ambientes deste tipo são denominadas *impressões digitais latentes* (GARRIS et al., 2001). A qualidade de uma impressão digital latente é bem inferior a de uma impressão digital previamente coletada. Tipicamente, somente uma porção da impressão está presente, deixando o resultado final incompleto além de muitas vezes aparecer borrado (GARRIS et al., 2001). A figura 3.2 mostra uma impressão digital coletada em ambiente controlado (figura 3.2a) e uma impressão digital latente (figura 3.2b).

Impressões digitais latentes são muito difíceis para sistemas automatizados de identificação funcionarem com segurança e confiabilidade.

Independente de se utilizar os processos de identificação ou verificação em uma impressão digital coletada em ambiente controlado ou latente, uma coisa é certa: a extração das características

¹Conjunto de características extraídas de uma impressão digital.

Figura 3.2: Impressões digitais: a) coletada em ambiente controlado b) latente

é baseada nas minúcias de uma impressão digital. Este trabalho se baseia nas características contidas em minúcias previamente localizadas. Não há o interesse de se desenvolver algoritmos novos para a detecção das minúcias e sim utilizar um já amplamente testado na literatura. Para isso utilizou-se o software *mindtct* desenvolvido pelo *National Institute of Standards and Technology* (NIST) explicado na próxima seção.

3.2 O software *Mindtct*

O *mindtct* é um software para detecção de minúcias e faz parte do pacote *NIST Fingerprint Image Software* (NFIS) desenvolvido pelo NIST. Seu uso é livre e com código aberto implementado na linguagem C. Sua obtenção dá-se através de contato com seus desenvolvedores².

Seu funcionamento é através de linha de comando onde deve-se passar um arquivo de formato proprietário do NIST contendo a imagem da impressão digital e como saída tem-se a imagem

²O endereço eletrônico para contato pode ser encontrado no site <http://fingerprint.nist.gov>.

de entrada acrescentada das minúcias detectadas. Um exemplo do resultado de detecção de minúcias feito pelo software pode ser visto na figura 3.3.

(a) Impressão Digital de Entrada

(b) Minúcias detectadas

Figura 3.3: Minúcias detectadas pelo software *mindtct*

O software foi dividido em fases como mostra a figura 3.4. Esta divisão em fases ajuda na modularização do sistema, visto que podem ser feitas melhorias nos algoritmos destas sem afetar o funcionamento e a estabilidade das demais. Nas subseções a seguir serão comentadas resumidamente as fases do *cálculo do mapa direcional*, *binarização da imagem*, *detecção de minúcias* e *contagem de cristas vizinhas*.

3.2.1 Mapa Direcional

Assim como nos trabalhos de (ROSS; JAIN; REISMAN, 2003; KAWAGOE; TOJO, 1984; HONG; WAN; JAIN, 1998), o *mindtct* se baseia no mapa direcional da impressão digital para o processo de detecção de minúcias. O propósito deste mapa é representar as cristas da impressão digital sem a presença de ruídos. Cristas bem formadas e livres de ruídos são essenciais para a detecção

Figura 3.4: Fases do processo de detecção de minúcias.

confiável das minúcias. Em adição, o mapa direcional registra a orientação geral do fluxo das cristas pela imagem (GARRIS et al., 2001).

Para analisar a impressão digital, a imagem é dividida em blocos. Todos os pixels dentro de um bloco são assinalados com os mesmos resultados. Assim, no caso do mapa direcional, todos os

pixels em um bloco serão assinalados com o mesmo fluxo de direção da crista.

Na figura 3.5, uma imagem original de impressão digital é mostrada na imagem à esquerda (figura 3.5a). A imagem à direita, é a mesma impressão digital tendo seu mapa direcional sobreposto (figura 3.5b). Cada direção no mapa é representado como um segmento de linha centrado dentro de um bloco composto por 8x8 pixels.

Figura 3.5: Mapa direcional de uma impressão digital

3.2.2 Binarização da Imagem

O algoritmo de detecção de minúcias opera em imagem de cores binarizadas, nos quais os pixels pretos representam as cristas e os pixels brancos os vales de uma impressão digital. Para criar a imagem binária, todos os pixels da imagem de entrada devem ser analisados para determinar se serão assinalados como preto ou branco. Este processo é denominado binarização.

A um pixel é assinalado um valor binário baseado no fluxo direcional das cristas associado com o bloco que o pixel pertence. Se não for detectado o fluxo da crista para o bloco de pixel

corrente, então o pixel é estabelecido como branco. Se for detectado o fluxo de crista, então os pixels em torno do pixel corrente são analisados dentro de um *grid* rotacionado. Este *grid* possui altura de nove pixels e comprimento de sete pixels. O processo funciona colocando o pixel de interesse no centro do *grid* que, então, é rotacionado até que suas linhas fiquem em paralelo com o fluxo direcional da crista. As intensidades de cinza dos pixels são acumuladas ao longo de cada linha rotacionada formando um vetor de somatório de linhas. O valor binário que será assinalado no pixel central é determinado multiplicando a somatória da linha central com o número de linhas no *grid* e comparando este valor com as intensidades de cinzas acumuladas dentro do *grid* inteiro. Se a multiplicação da linha central é menor que a intensidade total do *grid*, então o pixel central é assinalado como preto, senão, é assinalado como branco. O resultado do processo de binarização é mostrado na figura 3.6. Tem-se na figura 3.6a a imagem em escala de cinza e na figura 3.6b, a imagem binarizada.

(a)

(b)

Figura 3.6: Resultado da binarização

3.2.3 Detecção de Minúcias

Nesta etapa, a imagem binarizada da impressão digital é varrida a procura de padrões de pixels que indiquem um término ou uma bifurcação da crista. A varredura por padrões é conduzida tanto verticalmente quanto horizontalmente. Os padrões procurados são ilustrados na figura 3.7. Existem dois padrões representando cristas finais e os demais representando cristas bifurcadas. Todos os pixels são comparados com estes padrões formando uma lista de minúcias candidatas.

Figura 3.7: Padrões usados para detecção de minúcias

Usando os padrões da figura 3.7, as minúcias candidatas são detectadas em um bloco de seis pixels. Isso garante que nenhuma minúcia real será deixada fora da lista, mas também pode fazer que surjam falsas minúcias. Uma série de algoritmos é empregada para a remoção de ilhas, lagos, buracos, minúcias em regiões da imagem de baixa qualidade, ganchos e minúcias muito extensas ou muito curtas. Maiores detalhes podem ser encontrados na referência (GARRIS et al., 2001).

3.2.4 Contagem de Minúcias Vizinhas

Detectores de minúcias fornecem informações adicionais além das minúcias propriamente ditas. Entre estas informações têm-se as coordenadas X e Y , seu tipo, ângulo - esta característica também é denominada de direção e é representada pela letra grega teta (Θ). Na figura 3.8 são visualizadas estas características. Além da posição, ângulo e o tipo, não existe um padrão para informações de vizinhança de minúcia. Diferentes sistemas usam diferentes topologias e atributos para vizinhança de minúcia. O software *mindtct* utiliza o padrão adotado pelo FBI que identifica as oito minúcias vizinhas mais próximas e o número de cristas existentes entre a minúcia e sua vizinha.

Figura 3.8: Características de uma minúcia extraídas por detectores de minúcias

3.3 Pesquisas com Impressões Digitais

Como já citado, este trabalho aborda duas áreas: reconhecimento e indexação de impressões digitais. A seguir são explanadas algumas pesquisas recentes envolvendo as áreas de interesse

desta tese.

3.3.1 Algoritmos de Identificação de Impressões Digitais

Allan S. Bozorth (GARRIS et al., 2002) desenvolveu um algoritmo batizado de *Bozorth Matcher* baseado em minúcias cujo propósito é realizar identificação ou verificação de impressões digitais. As características das minúcias usadas são as coordenadas x,y e sua direção (θ). O funcionamento do algoritmo é baseado na construção das tabelas de compatibilidade de minúcias e de impressões digitais e a busca por padrões nesta última tabela. O primeiro passo do algoritmo é criar a tabela de comparação *Intra-Fingerprint Minutia*. Nesta tabela são colocadas as medidas relativas (*relative measurements*) de cada minúcia com todas as outras minúcias da impressão digital. Cada vetor de característica da tabela é formado por $\{d_{ij}, \beta_i, \beta_j, \Theta_{ij}, i, j\}$, com d_{ij} indicando a distância relativa entre duas minúcias. β_i e β_j medem o ângulo relativo da minúcia com respeito à linha de conexão destas minúcias. Θ_{ij} define a direção da linha de conexão e i e j são as posições das minúcias na impressão digital. Em seguida é construída a tabela de compatibilidade entre impressões digitais (*Inter-Fingerprint Compatibility*). Nesta tabela são colocados os vetores "compatíveis" das impressões comparadas. Por último, o grafo de compatibilidade é construído pela travessia na tabela de compatibilidade. Desta travessia, é gerada a pontuação que indica o quanto as impressões digitais são iguais.

Chikkerur et al (CHIKKERUR; CARTWRIGHT; GOVINDARAJU, 2005) propõem um algoritmo de reconhecimento de impressões digitais baseado em grafos denominado de *K-plet*. Ele consiste de uma minúcia central m_i e outras K minúcias $\{m_1, m_2, \dots, m_k\}$ escolhidas de sua vizinhança. Cada minúcia vizinha possui as características $(\Phi_{ij}, \Theta_{ij}, r_{ij})$. r_{ij} representa a distância euclidiana entre a minúcia m_i e m_j . Φ_{ij} é a orientação relativa da minúcia m_j em relação minúcia central m_i . Θ_{ij} representa a direção da conexão entre as duas minúcias e é também a medida relativa para orientação da minúcia m_i . A busca é realizada por um algoritmo *Coupled Breadth First Search (CBFS)*. A busca consiste em encontrar *K-plet* semelhantes nas impressões digitais

comparadas.

Wang e Gavrilova (WANG; GAVRILOVA, 2006) desenvolveram um algoritmo de reconhecimento cujo o vetor de características é $\{x, y, \Theta, t, \Delta length, \Delta\Theta_1, \Delta\Theta_2, \Delta t_1, \Delta t_2, \Delta rc\}$, onde $(x, y, \Theta$ e $t)$ são as coordenadas x e y da minúcia, Θ a direção da minúcia e t é o tipo dela. O algoritmo faz o uso da triangulação *Delaunay*. As característica $\Delta length$, refere-se ao comprimento dos lados do triângulo. $\Delta\Theta_1$ é os ângulo entre os lados e o campo de orientação da primeira minúcia. Δt_1 indica o tipo da primeira minúcia e Δrc é a quantidade de cristas entre os pontos de cruzamento das duas minúcias. A busca usando a triangulação *Delaunay* é feita da seguinte maneira: Se um dos lados do triângulo da imagem de entrada combina com dois lados de triângulos na imagem armazenada, é necessário considerar a triangulação para o qual este lado pertence e comparar com seu respectivo triângulo. Para uma faixa de translação e rotação, é detectado um pico dentro de um espaço de transformação e armazenadas as transformações que são vizinhas deste pico no espaço de transformação. A combinação destas características determina a busca pela triangulação *Delaunay*. Além disto, para minimizar os problemas causados por deformação, é usado o método *Radial Basis Functions* (RBF). A aplicação deste método permite realizar um alinhamento em ambas as imagem. Após o alinhamento, as minúcias que estiverem alinhadas e possuírem a mesma direção são consideradas iguais.

3.3.2 Indexação de Impressões Digitais

Além das três áreas de maiores pesquisas em impressão digital - classificação, detecção de minúcias e reconhecimento - citadas previamente na seção 3.1 deste capítulo, uma outra área que tem ganhado importância é o processo de indexação em banco de dados de impressões digitais. A importância se dá pelo fato que reconhecer uma impressão digital de uma base pequena é um trabalho onde o problema envolvido recai sobre a qualidade do algoritmo em determinar se duas impressões digitais são semelhantes ou não. Entretanto, quando se tem uma base grande de impressões digitais, outro problema surge: *o tempo de localização*.

Tradicionalmente, existem três tipos de abordagens para resolver o processo de reconhecimento de impressões digitais:

- Repetir o processo de verificação para cada impressão digital do banco de dados;
- Classificar as impressões digitais e
- Indexar as impressões digitais.

Se o tamanho do banco de dados for grande, a primeira abordagem se torna impraticável. No processo de classificação, as impressões digitais são separadas em cinco classes: *Right Loop* (R), *Left Loop* (L), *Whorl* (W), *Arch* (A) e *Tented Arch* (T). Entretanto, o problema com a técnica é que o número das classes é pequeno e a distribuição das impressões digitais entre estas classes não é uniforme. Estima-se que 31,7%, 33,8%, 27,9%, 3,7% e 2,9% pertençam, respectivamente, às classes R, L, W, A e T. A abordagem de classificação não estreita a pesquisa suficientemente no banco de dados para uma identificação eficiente de uma impressão digital. Assim, a abordagem de indexação pode ser considerada uma solução para o processo de reconhecimento de impressões digitais quando se tem um banco de dados muito grande.

O processo de indexação de impressões digitais se baseia em características encontradas na própria impressão digital. Na literatura são encontrados três métodos de indexação baseados nas características de *directional field*, *fingercodes* (JAIN; PRABHAKAR; HONG, 1999) e *minutiae triplets* (BHANU; TAN, 2003).

O *directional field* (DF) descreve o formato da impressão digital. Indica o fluxo que as cristas seguem. Para obter o vetor de característica, o DF é calculado por blocos de tamanhos fixos, por exemplo 16x16 pixels. Os vetores resultantes dos blocos são concatenados para formar o único vetor de característica da impressão digital. A dimensão deste vetor é ainda reduzida pela aplicação de *principal component analysis* (PCA) (HAYKIN, 1999).

Fingercodes é um esquema de representação que captura as características global (crista, delta e núcleo) e local (crista e minúcia) de uma impressão digital em um vetor de comprimento fixo. A

comparação é realizada aplicando a distância euclidiana entre os *fingercode*s de entrada e os armazenados no banco de dados. Trabalhos relacionados com *fingercode* podem ser encontrados em (ROSS; JAIN; REISMAN, 2003) , (JAIN et al., 2000).

Uma abordagem inicial envolvendo a indexação de impressões digitais foi realizado por Germain e outros (GERMAIN; CALIFANO; COLVILLE, 1997). Os autores usaram um conceito denominado *triplets of minutiae* como procedimento de indexação. As características usadas são o comprimento de cada lado (representado pela letra *S* na figura 3.9), o número de cristas entre cada par de vértices e os ângulos que as cristas fazem com o eixo *X* referente (representado pela letra grega θ da figura 3.9). Entretanto, esta abordagem possui alguns problemas: o número de cristas é sensível à qualidade da imagem, os ângulos formados pelas cristas e seu eixo *X* também são sensíveis à qualidade da imagem. Uma melhoria desta abordagem é encontrada em (BHANU; TAN, 2003), onde os autores, além de utilizarem o *triplets of minutiae*, utilizaram também os ângulos do triângulo formado, o tipo da minúcia, a direção e o maior lado do triângulo. Com o uso dessas características, os autores resolveram alguns problemas da proposta original.

3.4 Considerações Finais

Neste capítulo foram apresentados os principais conceitos envolvendo pesquisas com impressões digitais. Uma vez que este trabalho propõe um algoritmo de reconhecimento de impressões digitais e a indexação da mesma usando distância métrica, o trabalho de detecção de minúcia foi deixado a cargo do programa *mindtct* que foi desenvolvido pelo NIST. Além das tradicionais áreas de pesquisas em impressões digitais (classificação, detecção de minúcias e verificação), vem surgindo interesse na pesquisa para indexação das impressões digitais em grandes bancos de dados. Neste tipo de pesquisa surgem dois problemas: i) um método para o reconhecimento das impressões digitais e ii) um método para indexá-las. Este trabalho propõe

Figura 3.9: *Triplet of minutiae*: (a) Um *triplet* sobre uma imagem de impressão digital afinada. (b) Um *triplet* formando um triângulo sem a imagem da impressão digital

soluções que serão discutidas no capítulo referente à metodologia proposta.

Capítulo 4

Recuperação de Imagens Baseada em Conteúdo

4.1 Introdução

Imagens digitais são uma classe de dados que a cada dia vem se tornando ainda mais importante, principalmente após o aumento da capacidade de processamento e de memória dos computadores. São diversas áreas que produzem e utilizam imagens digitais, tais como medicina, militar, segurança, meteorológica, etc. Como a demanda por imagens digitais aumenta, torna-se necessário o desenvolvimento de pesquisas com intuito de obter melhores métodos de armazenagem e recuperação destas.

O campo de CBIR, sigla para *Content-Based Image Retrieval* ou Recuperação de Imagem Baseada em Conteúdo, focaliza métodos para recuperação eficiente de imagens, baseado em informações contidas nelas próprias. Entre as informações utilizadas, tem-se cor, textura, forma e relacionamento espacial.

4.2 Contextualização de CBIR

Imagens eletrônicas estão sendo geradas de forma muito intensa por fontes de informações como satélites, sistemas biométricos, experiências científicas, sistemas biomédicos, etc. Para utilizar eficientemente estas informações, é necessário um sistema eficiente de CBIR. Este sistema auxilia os usuários a recuperar e manusear as imagens baseadas em seu próprio conteúdo. As áreas onde estas técnicas podem ser aplicadas são numerosas, destacando-se gerenciamento de museus e galerias de artes, sensoriamento remoto e gerenciamento de recurso terrestres, gerenciamento de banco de dados científicos, etc.

As abordagens iniciais para o uso de CBIR caminhavam em duas direções (GUDIVADA; RAGHAVAN, 1995). Na primeira, os conteúdos das imagens são modelados com um conjunto de atributos extraídos manualmente e gerenciados dentro de um *framework* de sistema de gerenciamento de banco de dados convencional. As consultas são realizadas usando estes atributos.

A segunda abordagem é baseada no uso de um sistema integrado de extração de características e reconhecimento de objeto para superar as limitações da recuperação baseada em atributos. Este sistema, automatiza a tarefa de extração das características e reconhecimento de objetos no momento que a imagem é inserida no banco de dados. Entretanto, esta abordagem é computacionalmente cara, difícil e tende a ser de um domínio específico de conhecimento.

Pesquisas recentes no campo de CBIR reconhecem a necessidade de haver uma sinergia entre estas duas abordagens. Como exemplo das áreas que reforçam esta idéia tem-se sistemas baseados em conhecimento, ciência cognitiva, processamento de imagem, reconhecimento de padrões, sistema de recuperação de informações, etc. Assim, a confluência de idéias tem culminado na introdução de representações originais de imagens e modelos de dados, algoritmos de processamento de consultas robustos e eficientes, interfaces de consultas inteligentes entre outros.

4.2.1 Níveis de consultas

Existem duas categorias de características usadas para realização de consultas (GUDIVADA; RAGHAVAN, 1995; ZACHARY; IYENGAR; BARTHEN, 2001): a primitiva e a lógica . Estas duas categorias agregam consultas com diferentes níveis de complexidades:

- **Consultas de nível 1:** As consultas de nível 1 envolvem as características denominadas *primitivas* como cor, forma e textura. Este tipo de consulta é objetiva e composta de características originadas diretamente de imagens usadas nos algoritmos de processamento de imagens. Exemplos desse tipo de consulta incluem "recupere todas as imagens com círculos vermelhos no centro da imagem" , "recupere imagens que contenham quadrados azuis, retângulos e losângulos" e "recupere as imagens semelhantes a esta". Este tipo de consulta foi posteriormente denominada de *Query By Example* (QBE) ou consulta por exemplo. Exemplo deste tipo de consulta é mostrado na figura 4.1.
- **Consultas de nível 2:** As consultas de nível 2 envolvem as características denominadas *lógicas* que requerem algum nível de inferência sobre a identidade das coisas na imagem. Uma base de conhecimento externa é necessária para este tipo de consulta. Consultas classificadas como nível 2 são aquelas consultas no qual se deseja encontrar objetos ou pessoas dentro das imagens. Um exemplo deste tipo de consulta pode ser observado na figura 4.2.
- **Consultas de nível 3:** Consultas deste tipo são compostas de idéias abstratas e atributos e requerem uma quantidade significativa de argumentos de alto nível a respeito do significado e proposta da consulta. Este tipo de consulta é muito difícil de se automatizar. O motivo é a ligação entre o conteúdo da imagem e o conceito abstrato necessário. Uma consulta deste tipo pode ser observada na figura 4.3.

Figura 4.1: Exemplo de consultas de nível 1 para a imagem da Lena (ZACHARY; IYENGAR; BARTHEN, 2001)

Figura 4.2: Exemplo de consultas de nível 2 para a imagem da Lena (ZACHARY; IYENGAR; BARTHEN, 2001)

Figura 4.3: Exemplo de consultas de nível 3 para a imagem da Lena (ZACHARY; IYENGAR; BARTHEN, 2001).

Consultas de nível 1 são consideradas como foco principal de pesquisas e desenvolvimento de sistemas em CBIR. As consultas de níveis 2 e 3 são difíceis de implementar, já que exigem um nível de abstração maior e geralmente necessitam de fontes externas de informações para auxiliar na recuperação das imagens. Devido possuir esta dificuldade para recuperação, são consideradas de *recuperação semântica de imagem*, uma subcategoria de CBIR (GUDIVADA; RAGHAVAN, 1995; ZACHARY; IYENGAR; BARTHEN, 2001). A diferença entre consultas de nível 1, envolvendo os atributos físicos das imagens e consultas de nível 2, envolvendo atributos lógicos das imagens, é denominada de *semantic gap*, ou diferença semântica.

4.2.2 Pesquisas em CBIR

De longe, os atributos *cor e textura* são os atributos mais usados em sistemas de CBIR (ZACHARY; IYENGAR; BARTHEN, 2001; SAHA; DAS; CHANDA, 2004; ZHANG, 2004). Entretanto, atualmente os sistemas de CBIR estão preocupados em diminuir a diferença semântica envolvida nas consultas realizadas pelos usuários através da combinação de atributos físicos das imagens. Na área de CBIR, encontram-se diversos trabalhos envolvendo os atributos *cor e texturas*. Trabalhos envolvendo esta união de atributos podem ser vistos em (LIU et al., 2005; SAHA; DAS; CHANDA, 2004; BRAHMI; ZIOU, 2004; ZHANG, 2004). Em (LIU et al., 2005) os autores propõem um método para reduzir a diferença semântica das consultas. O método separa as cores de uma imagens usando o padrão HSV (*Hue, Saturation e Value*). Cada cor detectada é classificada dentro de 10 cores possíveis. Já a saturação e o valor são distribuídos em 4 adjetivos indicando a saturação e a luminância da cor. Estes adjetivos são *pálido* e *puro* para saturação e *escuro* e *claro* para valor. No trabalho desenvolvido por (SAHA; DAS; CHANDA, 2004) os autores utilizam uma matriz de co-ocorrência de textura para descrever a textura de uma imagem. É criado um índice *fuzzy* de cores para prover melhor desempenho de recuperação nas consultas efetuadas pelos usuários. Em (BRAHMI; ZIOU, 2004) é explorada a combinação de abordagens

baseadas em metadados (como descrição textual, palavras chaves ou textos de livre forma) e as baseadas no conteúdo dos atributos da própria imagem (como cor e textura) para recuperação de imagens com a ajuda do usuário superando as desvantagens destes métodos quando usados de forma isolada. O sistema desenvolvido pelos autores permite a interação entre usuário e sistema no sentido de obter um *feedback* do usuário em relação às imagens recuperadas. O *feedback* permite ao usuário a refinar uma consulta via especificação da relevância dos itens recuperados. O sistema irá retornar um conjunto de possíveis combinações e o usuário dará *feedback* informando a relevância ou não dos itens. Por fim, (ZHANG, 2004) propõe um método usando os atributos cor e textura nas imagens. As imagens são armazenadas no banco de dados indexadas por estes dois atributos. Durante o processo de recuperação das imagens, as imagens são primeiramente classificadas pelo atributo cor e colocadas em um *ranking* de classificação. Num segundo passo, as imagens pertencentes ao topo do *ranking* são reclassificadas de acordo com as características de sua textura. O *ranking* de cores é feito utilizando a técnica denominada *perceptually weighted histogram* (PWH) baseada no espaço de cor CIEL*u*v. Do resultado obtido, o usuário poderá fazer o uso ou não da característica textura.

4.3 Índices para Recuperação de Imagens

O principal objetivo da área de CBIR é a recuperação eficiente de imagens previamente armazenadas em um banco de dados (LI; SIMSKE, 2002; SMEULDERS et al., 2000). Porém, existem áreas que, além da eficiência, necessitam de rapidez na recuperação das imagens. Os índices têm o objetivo de acelerar as buscas de informações armazenadas no banco de dados. A organização dos dados em índices assemelha-se com a aparência de uma árvore, daí o uso da palavra *tree* usado na maioria dos nomes de índices. O uso de índices em CBIR pode ser dividido em três tipos de classes de técnicas (SMEULDERS et al., 2000): técnica de particionamento de espaço, de particionamento de dados e baseado em distância. Para cada uma destas classes, tem-se uma estrutura de índices que obtém-se melhor proveito para o tipo de dado trabalhado.

Dê acordo com Smeulders e outros (SMEULDERS et al., 2000), a classe de técnica baseada em particionamento de espaço faz uso da estrutura de índice denominada *K-D Tree*, já para o particionamento de dados obtém-se melhor desempenho com o uso da *R-Tree* e para a classe baseada em distância tem-se a *M-Tree*. Uma análise mais completa dos tipos de índices disponíveis para a indexação de banco de dados de imagens encontra-se no capítulo 5.

4.4 Considerações Finais

Neste capítulo foram apresentados conceitos relativos ao campo de Recuperação de Imagens Baseada em Conteúdo (CBIR). O interesse desta área é crescente devido ao grande volume de imagens armazenadas em banco de dados que foram geradas por diversos tipos de aplicações.

A busca de imagens, através de suas características, pode ser dividida em três níveis: nível 1, nível 2 e nível 3. Em cada nível de consulta há um aumento na abstração das informações de consulta e também se aumenta a dificuldade de serem implementadas estas consultas que, muitas vezes, não possuem todas as informações necessárias, necessitando a existência de uma fonte externa para auxiliar a consulta.

Neste trabalho, o processo de extração de características classifica-se no nível 1, pois as informações necessárias encontram-se presentes na própria imagem.

Capítulo 5

Indexação em Banco de Dados de Imagens

5.1 Introdução

Em muitas aplicações, imagens constituem a maioria dos dados adquiridos e processados. Por exemplo, em aplicações de sensoriamento remoto e astronomia, grandes quantidades de dados oriundos de imagens capturadas por estações de coletas espalhadas por diversos pontos são recebidos para processamento, análise e armazenamento. Na medicina, um grande número de imagens de várias modalidades (como tomografia computadorizada, ressonância magnética, etc) são produzidas diariamente e usadas para auxiliar a tomada de decisões sobre um diagnóstico (PETRAKIS; FALOUTSOS, 1995).

As imagens são armazenadas com a finalidade de serem recuperadas por consultas realizadas sobre elas através de seus próprios atributos. Para suportar essas consultas, uma imagem armazenada em um Banco de Dados de Imagem (BDI) deve ser previamente processada para extração de suas características. Estas características são usadas para busca no BDI e determinação de qual imagem satisfaz o critério de seleção da consulta. A eficácia de um sistema de BDI depende da representação correta dos tipos dos conteúdos extraídos, os tipos de consultas que são permitidas nas imagens e das técnicas de buscas implementadas. Pode-se acrescentar a

estes fatores de sucesso a rapidez com que as buscas são realizadas. Usualmente, existem dois grupos de métodos para a indexação de imagens que são o Método de Acesso Multidimensional ou Método de Acesso Espacial e o Método de Acesso Métrico. O primeiro focaliza as buscas em vetores de coordenadas X e Y do objeto. O segundo é utilizado para indexar dados complexos através das distâncias entre os objetos.

5.2 Métodos de Acesso Espaciais

Os Métodos de Acesso Espaciais (MAEs) são usados para indexar objetos multidimensionais. Um objeto é considerado multidimensional se puder ser localizado por uma série de n coordenadas, sendo n maior que 1. Os MAEs partem do princípio que os dados manipulados pertencem ao domínio dos dados espaciais ou a um espaço de dimensão n , onde cada dimensão é representada por uma chave da relação.

Os MAEs são baseados em vetores de espaços de dimensão fixa, sendo que a localização de cada objeto pode ser mapeada por um vetor de valores (um em cada dimensão). A busca é realizada levando em conta a localização do objeto de busca dentro do vetor e se pertence ou não a uma região ou se está próximo o suficiente para poder ser retornado na consulta.

Existem diversas estruturas de árvores de acesso multidimensionais. Dentre elas pode-se citar a *Quad-Tree* (FINKEL; BENTLEY, 1974), a *K-D-Tree* (ROBINSON, 1981) e todas as variantes da *R-Tree* (GUTTMAN, 1984). Como esta última é largamente citada em diversos artigos e é usada como base de comparações para avaliar o desempenho de novas estruturas (SANTOS FILHO, R.; TRAINA, A.; TRAINA Jr. C.; FALOUTSOS C., 2001), ela será brevemente comentada a seguir.

R-Tree

A *R-Tree* (GUTTMAN, 1984) é uma hierarquia de objetos na forma de uma árvore balanceada inspirada na B-Tree (BAYER; MCCREIGHT, 1972). Cada nó da *R-Tree* contém um *array* de

entradas formado por (*chave*, *ponteiro*). Nos nós folhas da R-Tree, cada ponteiro é o identificador de um objeto, enquanto em um nó não folha, cada ponteiro aponta para o nó filho no nível abaixo. O número máximo de entradas de cada nó é determinado pelo *node capacity* ou *fan-out* e pode ser diferente para nós folhas e não folhas. Usualmente, a capacidade de um nó é determinada pelo tamanho da página do disco. Resultados experimentais em (OTTERMAN, 1992) apud (BOZKAYA; OZSOYOGLU, 1997; CHAVEZ et al., 2001) mostraram que as R-Trees tornam-se ineficientes para espaços de dimensões maiores que 20.

5.3 Métodos de Acesso Métricos

O conceito de busca por similaridade tem aplicações em diversos campos. Alguns exemplos são os banco de dados não convencionais, em que o conceito de busca por igualdade não é usado, e sim o conceito de similaridade de objetos, como imagens, impressões digitais ou sons, aprendizado de máquina e classificação no qual um novo elemento deve ser classificado de acordo com a semelhança de um elemento já existente; recuperação de texto (em que procura-se por palavras em um banco de dados de texto permitindo um pequeno número de erros), biologia computacional, em que deseja-se encontrar uma sequência de DNA ou proteína em um banco de dados permitindo que alguns erros ocorram devido a variações típicas dos objetos armazenados.

Todas estas aplicações têm características comuns. Existe um universo U de objetos e a uma função de distância não negativa $d : U \times U \rightarrow \mathbb{R}^+$ definida sobre eles e pertencente a um *espaço métrico*.

Um espaço métrico é uma coleção de objetos e uma função de distância definida sobre eles. Uma função de distância $d(x, y)$ para um espaço métrico deve possuir as seguintes propriedades (NAVARRO, 2002; CHAVEZ et al., 2001; BOZKAYA; OZSOYOGLU, 1997):

1. Simetria: $d(x, y) = d(y, x)$

2. Não negatividade: $0 < d(x, y) < \infty$ se $x \neq y$ e $d(x, x) = 0$

3. Desigualdade triangular: $d(x, y) \leq d(x, z) + d(z, y)$

No espaço métrico, nenhuma informação geométrica, como no espaço euclidiano, pode ser usada. Tem-se somente um conjunto de objetos e uma função de distância $d()$ que pode ser usada para computar a distância entre dois objetos quaisquer.

5.4 Consultas por similaridade

Existem basicamente dois tipos de consultas utilizados pelos MAMs (BOZKAYA; OZSOYOGLU, 1997; CHAVEZ et al., 2001):

- Consulta por Abrangência (*Range Query - RQ*): A consulta $RQ(o_q, r_q)$, recupera todos os elementos dentro de uma distância r_q (raio de busca), a partir do objeto de referência o_q (objeto de busca). Onde $o_q \in U$. Formalmente, a consulta é definida:

$$RQ(o_q, r_q) = \{u \in U | d(o_q, u) \leq r_q\}$$

A figura 5.1 ilustra uma consulta por abrangência. Nesta figura tem-se todos os objetos pertencentes ao universo U de objetos e o subconjuntos de objetos A , onde $A \subseteq U$, recuperados pela consulta $RQ(o_q, r_q)$, representados pelos círculos cinza.

Figura 5.1: Exemplo de *Consulta por Abrangência*: A figura ilustra uma consulta por abrangência - $RQ(o_q, r_q)$ em um espaço métrico bidimensional utilizando uma função Euclidiana E_1 . O objeto o_q é o objeto de busca enquanto os objetos cinza constituem os objetos do conjunto resposta A .

- Consulta K-Vizinhos Mais Próximos (*k-Nearest Neighbor Query* - $kNNQ$): $kNNQ(q, k)$ é uma consulta aos k -vizinhos mais próximos que visa recuperar os k objetos mais próximos ao objeto de referência o_q , no qual $o_q \in U$. Formalmente, pretende-se encontrar o subconjunto $A \subseteq U$ que atenda a:

$$kNNQ(o_q, k) = \{\forall u \in A, v \in U - A, d(o_q, u) \leq d(o_q, v), |A| = k\}$$

A figura 5.2 ilustra uma Consulta aos k -Vizinhos Mais Próximos - $kNNQ(o_q, 4)$, neste caso, os quatro mais próximos. Nesta figura, tem-se o universo U de objetos e o subconjunto A , sendo $A \subseteq U$, contendo os quatro objetos mais próximos (representados por círculos cinzas) do objeto o_q recuperados pela consulta $kNNQ(o_q, 4)$.

Figura 5.2: Exemplo de Consulta aos k -Vizinhos Mais Próximos - $kNNQ(o_q, 4)$ em um espaço bidimensional utilizando a função de distância euclidiana E_1 como a função de distância. O objeto o_q é o objeto de busca enquanto os objetos cinzas constituem os objetos do conjunto resposta A.

5.5 Indexação de Dados em Domínio Métrico

A desigualdade triangular de um espaço métrico pode ser usada para descartar ou aceitar agrupamentos de objetos comparando o objeto utilizado na consulta com o representante do agrupamento. Isto é obtido inserindo uma função de distância métrica dentro de uma estrutura de índice.

Índices usados para consulta em banco de dados normalmente utilizam uma estrutura similar a uma árvore. Essa analogia do formato da estrutura do índice com árvore surgiu originalmente do trabalho de Bayer (BAYER; MCCREIGHT, 1972), quando os autores propuseram a estrutura de indexação denominada *B-Tree*. Índices que utilizam função de distância como fator de indexação são denominados de *árvores métricas*. Em uma árvore métrica, cada nó representa um agrupamento de dados e é descrito por um elemento representante (chamado de centro do agrupamento) e um raio do agrupamento definido pela distância máxima entre o centro e um elemento do agrupamento. Sempre que um agrupamento resultante de uma consulta por similaridade possuir um raio muito grande, a consulta é recursivamente refinada em agrupamentos

menores.

Existem diversas estruturas de MAMs, dentre as quais pode-se citar a *BK-Tree*, *VP-Tree*, *GH-Tree* e a *M-Tree* entre outras.

BK-Tree

A BK-Tree foi proposta em (BUKHARD; KELLER, 1973). Os autores propuseram uma estrutura apropriada para funções de distância com uso de valores discretos. No topo da árvore é escolhido um elemento arbitrário de um domínio de chaves e é feito um agrupamento com o resto das chaves com suas respectivas distâncias a partir da chave representativa. As chaves que estão na mesma distância da chave representativa, são colocadas no mesmo grupo. A mesma composição hierárquica é feita sobre todos os grupos recursivamente, criando uma estrutura de árvore.

VP-Tree

Proposta por (UHLMANN, 1991) a *Vantage-Point Tree* ou VP-Tree é uma árvore que basicamente particiona os dados em torno de um *Vantage Point* (VP) ou objeto representante. Este particionamento é semelhante ao que ocorre em árvores binárias de buscas. É escolhido um elemento qualquer para a raiz da árvore e é calculada a mediana do conjunto de todas as distâncias, $M = \text{mediana}\{d(p,u) / u \in U\}$. Os elementos u cujo $d(p,u) \leq M$ são inseridos dentro da subárvore esquerda, enquanto os elementos cujo $d(p,u) > M$ são inseridos na subárvore direita. O custo de construção da árvore é $O(n \log n)$ no pior caso, desde que ela esteja balanceada.

GH-Tree

Outra proposta de Uhlmann (UHLMANN, 1991) é a "*Generalized-Hyperplane Tree*" (GHT). É uma árvore binária. Em cada nó escolhem-se dois elementos denominados de pivô 1 ($p1$) e pivô

2 (p_2). Os elementos próximos à p_1 são colocados na subárvore da esquerda e os elementos próximos à p_2 são colocados na subárvore da direita. As consultas são realizadas avaliando $r_1 = d(q, p_1)$ e $r_2 = d(q, p_2)$. Se $r_1 - r < r_2 + r$ então a consulta entra na subárvore à esquerda. Se $r_2 - r \leq r_1 + r$ então a consulta entra na subárvore à direita.

M-Tree

Na *M-Tree* todos os objetos são armazenados apenas nas folhas e a intersecção das regiões do espaço métrico, abrangidas por duas subárvores de um mesmo nó, pode não ser vazia. Ela armazena um dado conjunto de objetos $\{o_1, \dots, o_n\}$ em nós folhas de tamanho fixo, que correspondem às regiões do espaço métrico. Cada entrada do nó folha contém o identificador do objeto, os dados para cálculo de distância, e a distância ao objeto representante, o qual também é armazenado no nó pai. Cada entrada do nó interno armazena um ponteiro para a sua subárvore, um objeto representante, sua distância ao objeto representante pai e o raio da região que armazena os objetos indexados. Nas buscas, os objetos são descartados pelo uso do raio de cobertura e das distâncias dos objetos a seus representantes.

5.5.1 A Slim-Tree

A árvore métrica *Slim-Tree* foi criada por (TRAINA Jr., C.; TRAINA, A.; SEEGER, B.; FALOUTSOS C., 2000). Ela, diferentemente de outras MAMs como a *VP-Tree* (YIANILOS, 1993), *GH-Tree* (UHLMANN, 1991) e *MVP-Tree* (BOZKAYA; OZSOYOGLU, 1997) entre outras, não é estática. Isto significa que permite que novos objetos sejam inseridos e retirados dinamicamente. A estrutura de dados da *Slim-Tree* segue a mesma idéia de outras árvores métricas, como a *M-Tree* (CIACCIA; PATELLA; ZEZULA, 1997), na qual os dados são inseridos nas folhas e o balanceamento é feito pela altura. Outra semelhança é que a intersecção dos subespaços métricos definidos pelos nós de um mesmo nível pode não ser vazia. Ou seja, a divisão do espaço métrico não gera regiões disjuntas.

A *Slim-Tree* possui um recurso denominado de *Fat-Factor* que permite avaliar o grau de sobreposição entre seus nós e um algoritmo de otimização da árvore chamado de *Slim-Down*. Como outras árvores de métricas, os objetos do conjunto de dados são agrupados dentro de páginas de discos de tamanho fixo, sendo que cada página representa um nó da árvore.

Os nós da *Slim-Tree* podem ser de dois tipos: (i) *nó índice*, usado para guiar a consulta até aos objetos e (ii) *nó folha*, usado para armazenar os objetos. Os nós índices possuem cardinalidade igual ao seu grau. Todos os nós da estrutura, exceto o nó raiz, possuem um objeto representante e um raio de cobertura que engloba todos os objetos do nó e de suas subárvores.

Um nó folha tem a seguinte entrada:

$$\textit{leafnode} [\text{vetor de } \langle o_i, Old_i, d(O_{rep}, O_i) \rangle],$$

onde

- o_i representa o objeto;
- Old_i representa o identificador do objeto e
- $d(o_{rep}, o_i)$, representa o valor da distância entre este objeto e o representante.

Cada nó índice tem a seguinte entrada:

$$\textit{indexnode} [\text{vetor de } \langle o_i, d(o_{rep}, o_i), Ptr(To_i), R_i, NEnt(Ptr(To_i)) \rangle],$$

onde

- o_i representa o objeto representante subárvore;
- $d(o_{rep}, o_i)$, representa o valor da distância entre este objeto e o representante do nó;
- $Ptr(To_i)$, representa um ponteiro para sua subárvore;

- R_i , representa um raio de cobertura abrangendo toda subárvore e
- $NEnt(Ptr(To_i))$ que representa o número de entradas presentes no nó apontado por $Ptr(To_i)$.

5.5.1.1 Construindo uma *Slim-Tree*

A construção da *Slim-Tree* dá-se pela raiz até as folhas num processo conhecido como *bottom-up*, e uma vez que a *Slim-Tree* é uma árvore dinâmica, ela pode ser construída sem a necessidade da presença de todos os elementos que compõe a árvore. Os objetos podem ser inseridos mesmo após a estrutura estar construída.

Os objetos são inseridos em uma *Slim-Tree* seguindo um procedimento. Começando pelo nó raiz, o algoritmo tenta localizar um nó que possa guardar o novo objeto. Se nenhum se qualifica, o algoritmo seleciona o nó cujo centro está próximo do novo objeto. Se mais de um nó se qualifica, o algoritmo *ChooseSubtree* seleciona um deles. Este processo é recursivamente aplicado para todos os níveis da árvore.

A *Slim-Tree* tem três opções para o algoritmo *ChooseSubtree*:

- **Randômico** (*random*): que aleatoriamente escolhe um dos nós qualificados;
- **Distância Mínima** (*mindist*): escolhe o nó que tem a menor distância entre o objeto novo e o centro do nó;
- **Ocupação Mínima** (*minoccup*): escolhe, entre os nós qualificados, aquele cujo número de objetos armazenados seja mínimo. Isto é verificado pelo atributo *NEnt*.

Os algoritmos de redistribuição da *Slim-Tree* são:

- **Randômico** (*Random*): seleciona aleatoriamente entre objetos representantes, um para cada nó(novo e antigo) e os demais objetos são distribuídos entre os dois nós. Cada

objeto é armazenado em um novo nó de acordo com a proximidade do objeto em relação ao centro. A taxa de ocupação deve ser respeitada. É o algoritmo mais rápido, entretanto, seus resultados são os menos satisfatórios;

- **minMax:** todos pares de objetos possíveis são considerados candidatos a representantes dos dois nós. Para cada par, o algoritmo insere os objetos ao representante mais próximo. A complexidade do algoritmo é $O(C^3)$, onde C é a cardinalidade do nó, pois, se escolhe, para cada objeto dentre os demais quem será seu par e, para cada par possível, verifica-se em qual dos nós centrados em um dos elementos do par cada objeto restante será armazenado. Este algoritmo é também utilizado na *M-Tree* e é considerado o mais promissor algoritmo de realocação de árvores pois possibilita consultas mais eficientes.
- **MST** (*Minimal Spanning Tree* (KRUSKAL, 1956)): A árvore de caminho mínimo é construída e um dos arcos mais longos da árvore é removido. Este algoritmo produz *Slim-Trees* quase tão boas quanto as geradas pelo algoritmo minMax, porém com custo reduzido. Isto ocorre devido a complexidade do algoritmo ser $O(C^2 \log C)$. A figura 5.3 ilustra o funcionamento do algoritmo. Em (a) tem-se um nó contendo 8 objetos, sendo o objeto A o representante do nó; em (b) é mostrada a árvore de caminho mínimo para os objetos do nó corrente e em (c) é mostrada a distribuição para os dois novos nós.

5.5.1.2 Consultas por Similaridade na *Slim-Tree*

A *Slim-Tree* implementa dois algoritmos de consulta por similaridade: Consultas por Abrangência ($RQ()$) e os K -Vizinhos mais próximos ($KNNQ()$), citados na seção 5.4.

Independente do tipo de consulta, o algoritmo de busca inicia a busca pelo nó raiz comparando o objeto de busca com todos os objetos representantes do nível abaixo. Somente as subárvores que cobrirem possíveis respostas serão percorridas de maneira recursiva. Uma distância só é calculada quando não for possível descartar o objeto através da propriedade da desigualdade

Figura 5.3: Funcionamento do algoritmo MST: a) Nó antes de divisão; b) MST construída sobre os objetos do nó; c) Nó depois da divisão

triangular. Com isso, é possível minimizar o número de acessos a disco e de cálculo de distância necessários para responder às consultas.

Para as consultas do tipo $RQ()$ a *Slim*- o raio de cobertura e as distâncias pré-calculadas para descartar subárvores e objetos nas folhas através da propriedade de desigualdade triangular. A subárvore somente será percorrida se houver a intersecção da região definida pelo objeto representante da subárvore e a definida pelo objeto de busca e seu raio.

As consultas do tipo $KNNQ()$ têm abordagem semelhante. Ela é tratada como uma $RQ()$, inicialmente com raio infinito para preencher o conjunto resposta com K objetos. O raio da consulta vai diminuindo a medida que o objeto mais distante do conjunto resposta for sendo substituído por um menos distante em relação ao objeto de consulta.

5.5.1.3 Otimização de Sobreposição

A sobreposição de nós é algo indesejado, pois obriga a busca em profundidade em diversas subárvores para a localização dos objetos solicitados pelas consultas. Tem-se sobreposição quando em vetores de espaços duas entradas referem-se à quantidade de espaço comum coberta por ambas regiões, ou seja, há intersecção entre os objetos do vetor (TRAINA Jr., C.; TRAINA,

A.; SEEGER, B.; FALOUTSOS C., 2000).

Para diminuir a sobreposição dos nós e reorganizá-los, a *Slim-Tree* utiliza um mecanismo para verificar a porcentagem de sobreposição entre os nós, denominado de *Fat-Factor*, e um algoritmo de reorganização, denominado de *Slim-Down* (TRAINA Jr., C.; TRAINA, A.; SEEGER, B.; FALOUTSOS C., 2000).

O *Fat-Factor* é uma medida para avaliar o grau de sobreposição dos nós da árvore. Toda vez que o *Fat-Factor* atinge um limite definido, o algoritmo *Slim-Down* executa a reorganização dos objetos encontrados em regiões de sobreposição. Os objetos são transferidos de um nó para outro quando esta transferência diminui o raio do nó de origem sem aumentar o raio do nó destino, diminuindo o grau de sobreposição entre estes nós. Esse processo é realizado até que não seja mais possível realizar trocas de objetos entre os nós envolvidos, diminuindo a sobreposição dos nós e, conseqüentemente, melhorando o desempenho da árvore para responder consultas. É importante salientar que a execução do *Slim-Down* deve ser feita somente em último caso, pois a troca de objetos entre nós é uma operação custosa.

5.5.1.4 Desempenho da *Slim-Tree*

Em Traina Jr. e outros (TRAINA Jr., C.; TRAINA, A.; SEEGER, B.; FALOUTSOS C., 2000), foi realizado um conjunto de testes comparando a *Slim-Tree* com a *M-Tree* demonstrando a eficiência dos algoritmos MST e *Slim-Down*. Os parâmetros utilizados foram o tempo gasto para criação da árvore e o número de acessos ao disco para desempenho de consulta.

Os testes foram realizado sobre seis conjuntos de dados reais e sintéticos (tabela 5.1). Alguns possuíam métricas próprias e em outros foram usados a métrica L_2 .

Nos teste comparativos com a *M-Tree* foi utilizado o algoritmo de realocação *minMax*. Dos testes realizados nos seis conjuntos de dados, a *Slim-Tree* mostrou-se superior à *M-Tree* na maioria deles. Entretanto, para conjuntos de dados cuja sobreposição dos dados de entrada era

Tabela 5.1: Conjunto de dados usado nos testes de desempenho da *Slim-Tree* (TRAINA Jr., C.; TRAINA, A.; SEEGER, B.; FALOUTSOS C., 2000).

Conjunto de Dados	Num.Objetos	Dimensão	Métrica	Descrição
Uniform2D	10,000	2	L_2	Dados Distribuídos Uniformemente
Sierpinsky	9,841	2	L_2	Conjuntos de Dados de Fractais
MGCounty	15,559	2	L_2	Pontos de intersecção de estradas de Montgomery County - Maryland
EigenFaces	11,900	16	L_2	Vetor de faces do projeto Informedia (WACKTLAR et al., 1996)
FaceIT	1,056	desconhecido	FaceIT	Conjunto de dados construído por uma matriz de distância obtida do software FaceIT
EnglishWords	25,143	nenhum	L_{edit}	Palavras do dicionário da Língua Inglesa

pequena, as diferentes estratégias de inserção de ambas as árvores apresentaram desempenho similares.

5.6 Considerações Finais

Este capítulo apresentou dois grupos de métodos para a indexação de imagens que são o Método de Acesso Multidimensional, também conhecido como Método de Acesso Espacial (MAE) e o Método de Acesso Métrico (MAM). A diferença entre eles consiste em, no primeiro, os objetos indexados serem localizados pelas coordenadas espaciais e, no segundo, os objetos indexados são localizados pela distância, calculada por uma função de distância métrica, entre eles e seus respectivos objetos representantes.

As consultas que podem ser realizadas nos MAMs são as consultas por abrangência (RQ) e as consultas pelos k -vizinhos mais próximos (k NNQ). No primeiro tipo de consulta é indicada uma

distância e o método recupera todos os objetos que estiverem dentro do raio de cobertura e, no segundo tipo, é indicado um objeto referência e o método recupera os k objetos que estiverem mais próximos ao objeto indicado.

Das estruturas de indexação apresentadas, a *Slim-Tree* foi a mais detalhada. Ela é uma estrutura de indexação do tipo MAM e o interesse por ela se deu pelo motivo de usá-la neste trabalho como estrutura de indexação das impressões digitais.

Capítulo 6

Metodologia e Resultados

6.1 Introdução

Como os métodos tradicionais de identificação de pessoas possuem problemas como esquecimento, perda, roubo, etc. Este trabalho apresenta uma solução a estes problemas através do uso da biometria para o reconhecimento e autenticação dos indivíduos.

As propostas são de i) desenvolver um novo algoritmo para identificação de impressões digitais em espaço métrico e ii) usá-lo para indexar um banco de dados de impressão digital através de uma árvore de busca métrica.

6.2 Metodologia

O *Metric Fingerprint Identification System* (MFIS) é uma abordagem para identificação de impressões digitais baseada em um algoritmo original que opera em espaço métrico e organiza impressões digitais de forma indexada através da MAM *Slim-Tree*.

A metodologia adotada neste trabalho pode ser dividida em duas fases:

A fase 1 tem o objetivo de selecionar as características das impressões digitais para criação do vetor de características e armazená-lo em um banco indexado de impressões digitais.

A fase 2 tem o objetivo de realizar as buscas de impressões digitais no banco e retornar as impressões mais semelhantes.

A figura 6.1 mostra um diagrama de blocos genérico com as principais etapas de cada fase. A Fase 1 inicia-se com a extração das características de uma impressão digital. A extração é realizada pelo software *mindtct* e o resultado desta extração é gravado em um arquivo XML contendo as características necessárias para esta metodologia. O arquivo XML é importado para dentro do MFIS em forma de um vetor de característica. Este vetor, então, é armazenado em um banco de vetores de características indexado. A indexação é o agrupamento dos vetores mais semelhantes e é realizada pela árvore *Slim-Tree* baseada no algoritmo métrico proposto (ver seção 6.2.2). É ele quem determina o quanto dois vetores são semelhantes. A Fase 2 também inicia-se pela extração das características da impressão digital e criação de seu vetor. Este vetor será procurado no banco de vetores de características indexados. A busca é feita através das distâncias métricas obtidas entre o vetor de entrada e os vetores armazenados. Novamente, a distância métrica é calculada pelo algoritmo proposto. O Resultado é uma lista com os vetores de características mais semelhantes ao vetor de entrada.

Um diferencial importante e inovador neste trabalho é o fato de, uma vez que o algoritmo opera em espaço métrico, ele se utiliza das propriedades fundamentais deste espaço (ver seção 5.3) para reduzir o volume de impressões digitais comparadas. Considere a figura 6.2, nela vê-se uma representação de um grupo de impressões digitais organizada de forma indexada. Observa-se também, a raiz da árvore com duas impressões digitais representantes *idx* e *idy* (para este exemplo somente, a semelhança das impressões digitais se dá pela semelhança de seus nomes, assim a impressão digital *idxxx* é mais semelhante à *idx* do que à *idy*), um nível inferior contendo outras duas impressões digitais (*idxx* e *idxy*) semelhantes à da raiz *idx* no nível acima. Quando uma busca é realizada, ela se inicia pela raiz e percorrerá somente os

nós índices que possuem impressões digitais semelhantes à impressão digital de entrada até alcançar o nó folha. O que determina qual o nó índice é mais semelhante é a distância métrica entre o elemento representante e a impressão digital de entrada. Ao se localizar o nó desejado, todos os demais são descartados. Uma vez alcançado o nó folha em que estão localizadas as impressões digitais, a busca procede-se da seguinte forma: se a impressão digital de entrada for semelhante ao representante do nó, serão retornadas as k impressões digitais mais semelhantes a este representante e que já estão organizadas nesta ordem. Se a impressão digital de entrada for mais semelhante à uma outra, o procedimento de busca procurará no nó, as que mais se assemelham com a impressão digital de entrada.

Assim, a função da *Slim-Tree* é de realizar a indexação, armazenagem e a busca das impressões digitais, porém, como ela foi desenvolvida para manipular diversos tipos de objetos, é o algoritmo proposto que possibilita seu uso para manipulação de impressões digitais e com isso, reduz a quantidade de impressões digitais comparadas em uma busca. Isto o difere de outros algoritmos, que para fazer o reconhecimento de impressões digitais, deve comparar a impressão digital de entrada, com todas as cadastradas na base de dados.

6.2.1 Vetor de Características

O vetor de características é armazenado em um banco de vetores (neste texto, banco de vetores de características e banco de impressões digitais, tem o mesmo significado) gerenciado pela MAM Slim-tree. Este banco de vetores é indexado.

6.2.1.1 Definição das Características

Assim como nos métodos tradicionais de reconhecimento de impressões digitais, as características relevantes são as localizações das minúcias baseadas nas posições X e Y. Mais precisamente a localização das minúcias dos tipos crista final e crista bifurcada.

Figura 6.1: Diagrama da Metodologia Proposta

Este trabalho não tem como proposta desenvolver um algoritmo para detecção de minúcias em impressões digitais. Por este motivo, foi utilizado o *software mindtct* (lê-se *mindetect*) que é integrante do sistema NFIS desenvolvido pelo NIST (GARRIS et al., 2002). A escolha deste software foi devido a precisão dos resultados, superiores aos resultados de outros softwares detectores de minúcias testados, por ser de uso gratuito e possuir código aberto. Além da imagem com a localização das minúcias, o *software mindtct* também gera um arquivo texto contendo as minúcias e suas respectivas posições.

Por ter seu código disponível, foram realizadas alterações para que o mesmo gerasse, ao invés de um arquivo texto de formato proprietário, um arquivo no formato *Extensible Markup Language* (XML). A figura 6.3 mostra um arquivo XML gerado para nossa metodologia.

Figura 6.2: Representação da busca de impressões digitais em espaço métrico.

Para o arquivo XML foram exportadas somente as características necessárias, visto que o arquivo original possui diversas informações irrelevantes para este trabalho. Na figura 6.3 observam-se as seguintes *tags* representando as características:

- `<fingerprint></fingerprint>`: Inicia e termina uma impressão digital. O conteúdo dentro destas *tags* descrevem as características da impressão digital.
- `<minucia></minucia>`: Indica as características de uma minúcia. Cada minúcia terá uma seção como esta.
- `<fingernam></fingernam>`: Informa o nome do arquivo origem que se extraiu o arquivo XML. O arquivo de origem é uma imagem.
- `<x></x>`: Esta *tag* indica a posição *X* de uma minúcia.
- `<y></y>`: Esta *tag* indica a posição *Y* de uma minúcia.

```

<?xml version="1.0"?>
<fingerprint>
  <minúcia>
    <fingername>1_1.eft</fingername>
    <x> 26</x>
    <y> 89</y>
    <type>1</type>
    <angle> 2</angle>
    <nbx0> 36</nbx0>
    <nby0> 137</nby0>
    <nridcount0>3</nridcount0>
    <nbx1> 41</nbx1>
    <nby1> 140</nby1>
    <nridcount1>4</nridcount1>
    <nbx2> 50</nbx2>
    <nby2> 145</nby2>
    <nridcount2>3</nridcount2>
    <nbx3> 80</nbx3>
    <nby3> 160</nby3>
    <nridcount3>5</nridcount3>
    <nbx4> 87</nbx4>
    <nby4> 178</nby4>
    <nridcount4>6</nridcount4>
  </minúcia>
</fingerprint>

```

Figura 6.3: Formato de um arquivo XML representando uma impressão digital

- `<type></type>`: Informa qual o tipo da minúcia, sendo 0 para terminação e 1 para bifurcação.
- `<angle></angle>`: Informa a direção da minúcia.
- `<nbxN></nbxN>`: Informa a posição X dos vizinhos da minúcia. N varia de 0 a 4.
- `<nbyN></nbyN>`: Informa a posição Y dos vizinhos da minúcia. N varia de 0 a 4.
- `<nridcountN></nridcountN>`: Informa a quantidade de cristas entre a minúcia e seu vizinho.

6.2.1.2 Criação e Armazenamento do Vetor de Características

Ao ser importada do arquivo XML, cada impressão digital é representada por um vetor de característica. Este vetor foi elaborado para armazenar características de acordo com a metodologia de reconhecimento proposta no qual está baseada em pesos para características similares encontradas em impressões digitais comparadas.

Parte das características usadas são encontradas no arquivo XML. As características são: *coordenadas X e Y, tipo da minúcia, direção da mesma e a quantidade de cristas*. Entretanto, somente estas características não são suficientes para garantir a similaridade entre duas impressões nesta metodologia. Adotou-se, também, a vizinhança de minúcia como características de similaridade.

Quando uma minúcia é comparada, ela é denominada de *minúcia alvo* (MA) e suas vizinhas de *minúcias vizinhas* (MV), são extraídas e armazenadas para posteriormente serem comparadas na fase de verificação. As características da minúcias alvos são (x, y, θ, t) e das minúcias vizinhas como (x, y, θ, t, rc) , onde x representa a coordenada x , y a coordenada y , θ a direção, t o tipo e rc a quantidade de cristas.

Uma minúcia alvo deve ter cinco minúcias vizinhas (esta quantidade de vizinhos foi escolhida por apresentar a melhor relação entre a precisão e o tempo de busca nos testes realizados) formando um vetor de características $V = \{x_{ma}, y_{ma}, \theta_{ma}, t, x_{mv1}, y_{mv1}, \theta_{mv1}, t_{mv1}, rc_{mv1}, \dots, x_{mv5}, y_{mv5}, \theta_{mv5}, t_{mv5}, rc_{mv5}\}$ representado pela figura 6.4. Este vetor que será armazenado de forma indexada no banco de vetores de impressões digitais e está apto a ser usado no processo de identificação.

A vizinhança de uma minúcia é escolhida da seguinte maneira: em uma imagem, as minúcias possuem as coordenadas X e Y. O algoritmo percorre a imagem começando pela posição $X=0$ e $Y=0$, em seguida ele percorre toda as posições referente à coordenada Y. Terminada o percurso, ele percorrerá novamente as posições da coordenada Y, porém, com o valor da coordenada X em 1. Terminado novamente o percurso, a coordenada Y será outra vez percorrida, só que agora,

x_{ma}	y_{ma}	θ_{ma}	t_{ma}	x_{mv1}	y_{mv1}	θ_{mv1}	t_{mv1}	rc_{mv1}	...	x_{mv5}	y_{mv5}	θ_{mv5}	t_{mv5}	rc_{mv5}
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

onde:

ma, representa a minúcia alvo;

mv, representa as minúcias vizinhas;

x, representa a coordenada x;

y, representa a coordenada y;

rc, representa o número de cristas entre a minúcia

alvo e a minúcia vizinha;

t, representa o tipo da minúcia;

θ , representa a direção da minúcia.

Figura 6.4: Detalhamento do vetor de características

a coordenada X valerá 2. Assim, uma minúcia é considerada vizinha de uma minúcia alvo, se aquela possuir um valor de X maior di que esta ou se os valores de X forem iguais, o valor da coordenada Y da minúcia vizinha for maior do que a minúcia alvo.

Para exemplificar a escolha de vizinhança, considere a figura 6.5. Nela têm-se quatro minúcias E1, E2, B1 e B2 juntamente com suas respectivas coordenadas. As minúcias de letra E representam as do tipo terminação e as com a letra B representam as do tipo bifurcação. Assim, se E1 for a minúcia alvo, suas vizinhas serão E2, B1 e B2 (representadas pela seta tracejada). Se E2 for a minúcia alvo, ela terá como vizinhas B1 e B2 (representadas pela seta contínua) e finalmente, se B1 for uma minúcia alvo, ela terá somente B2 (representadas pela seta pontilhada) como vizinha. Uma vez que não existem mais minúcias com as coordenadas X e Y maiores do que as de B2, então, esta não possui vizinhas.

Uma vez que a MAM *Slim-Tree* indexa os objetos baseada na distância métrica entre eles, vetores criados a partir de imagens da mesma impressão digital tendem a possuir uma menor distância métrica e quanto menor for a distância, mais próximos serão agrupados fisicamente os objetos no banco de dados.

6.2.2 Algoritmo Proposto

O diagrama de blocos ilustrado na figura 6.1 mostra que na fase 2 é realizado o reconhecimento da impressão digital. Nesta etapa o MFIS irá reportar as impressões digitais no banco de dados

Figura 6.5: Vizinhança de uma minúcia.

mais parecidas com a impressão digital de entrada. Para que isso ocorra, é necessário que um índice de similaridade seja calculado. Este índice é obtido pela distância métrica das impressões digitais.

A distância métrica é uma pontuação obtida pela somatória de cada característica do vetor de características que são semelhantes em impressões digitais comparadas. Deve-se, no entanto, observar algumas considerações no que diz respeito à comparação de características de impressões digitais.

A comparação das características é um processo delicado no tocante ao posicionamento das minúcias dentro de uma impressão digital. Uma mesma impressão digital, obtida de imagens diferentes, pode conter uma variação de posicionamento de minúcias. Na figura 6.6 tem-se uma mesma impressão digital em imagens diferentes. Note que a figura 6.6 a, não apresenta a existência de um delta. Já na figura 6.6 b, percebe-se (indicado por um círculo) que o delta está

posicionado na parte baixa da imagem. Esta característica nas imagens de impressões digitais dificultam processos de verificação baseados em coordenadas X e Y.

(a) Imagem de uma impressão digital sem delta aparente

(b) Imagem de uma impressão digital com delta aparente

Figura 6.6: Impressão digital com deslocamento de posição

O algoritmo proposto não utiliza os valores das coordenadas X e Y, mas características calculadas, como a distância euclidiana entre as MAs e MVs e a triangulação obtida entre elas. A não comparação dos valores das coordenadas, permite uma certa robustez com relação a fragmentos obtidos da imagem original (ver resultados dos testes na seção 6.4.2). Além destas duas características, também são usadas as características de direção, tipo da minúcia e a quantidade de cristas entre a minúcia alvo e a minúcia vizinha.

Estas características foram denominadas de padrões e foram definidos dois tipos de padrões: i) padrões primários, como as características direção, tipo, número de cristas e ii) padrões calculados, como a distância euclidiana entre minúcias alvos e vizinhas e a triangulação destas. Para detalhes da triangulação, ver seção 6.2.2.1.

Um procedimento inovador neste algoritmo é que para cada padrão encontrado, uma "pontuação" é atribuída. Outra característica importante é de garantir que, encontrados três triângulos em comum nas impressões de busca e na do banco de dados, a procura é interrompida pois as

impressões digitais são consideradas similares.

No final do processo de comparação das impressões, a "pontuação" obtida é retirada de uma pontuação limite que se pode obter. O resultado desta subtração é que define o índice de similaridade entre as impressões digitais comparadas.

- 1 Para cada MA_i de V_i , faça:
- 2 Para cada MV_i de V_i , faça:
- 3 Para cada MA_b de V_b , faça:
- 4 Para cada MV_b de V_b , faça:
- 5 Calcular distância euclidiana entre V_i (MA_i, MV_i) e armazenar em $DistMAMV_i$.
- 6 Calcular distância euclidiana entre V_b (MA_b, MV_b) e armazenar em $DistMAMV_b$.
- 7 Se as características $V_i(DistMAMV_i, \theta_{mai}, rc_{mvi}) = V_b(DistMAMV_b, \theta_{mab}, rc_{mvb})$
- 8 Para cada MV_{i+1} de V_i , faça:
- 9 Para cada MV_{b+1} de V_b , faça:
- 10 Calcular o triângulo $\Delta_i = (MA_i, MV_i, MV_{i+1})$
- 11 Calcular o triângulo $\Delta_b = (MA_b, MV_b, MV_{b+1})$
- 12 Se $\Delta_i = \Delta_b$
- 13 Calcular distância euclidiana entre V_i (MA_i, MV_{i+1}) e armazenar em $DistMAMV_2i$.
- 14 Calcular distância euclidiana entre V_b (MA_b, MV_{b+1}) e armazenar em $DistMAMV_2b$.
- 15 Se as características $V_i(DistMAMV_2i, \theta_{mvi+1}, rc_{mvi+1}) = V_b(DistMAMV_2b, \theta_{mvb+1}, rc_{mvb+1})$
- 16 $AcumPont = AcumPont + Pontuação$
- 17 $AcumTriang = AcumTriang + 1$
- 18 Se ($AcumTriang \geq 3$) ou V_i chegar no término das minúcias, parar a busca.
- 19 De acordo com o total de triângulos faça:
- 20 Para um triângulo, atribua pontuação de peso 1 no acumulador;
- 21 Para dois triângulos, atribua pontuação de peso 2 no acumulador;
- 22 Para três triângulos, atribua pontuação de peso 3 no acumulador;

Figura 6.7: Algoritmo da Função Computacional de Distância Métrica

A figura 6.7, descreve a função computacional de distância métrica para identificação de impressões digitais desenvolvida para o MFIS. Inicialmente, é escolhida um minúcia alvo (MA_i) e uma minúcia vizinha (MV_i) do vetor da impressão digital de entrada (V_i). É calculada a distância euclidiana entre elas ($DistMAMV_i$). Considerando esta informação, a direção de MA_i

(θ_{mai}) e a quantidade de cristas entre a MV_i e MA_i (rc_{mvi}) são percorridas as minúcias alvos (MA_b) do vetor da impressão digital do banco de dados (V_b). Se for encontrado $V_i(DistMAMV_i, \theta_{mai}, rc_{mvi}) = V_b(DistMAMV_b, \theta_{mab}, rc_{mvb})$ é escolhida a próxima vizinha de MA_i (MV_{i+1}). Com as três minúcias (MA_i , MV_i e MV_{i+1}) é calculado a triangulação considerando-se os ângulos $\angle_{1i}$, $\angle_{2i}$ e $\angle_{3i}$ (representados no algoritmo por Δ_i). Este triângulo também é calculado com as minúcias vizinhas de MA_b (MA_b , MV_b e MV_{b+1}). O processo do cálculo da triangulação é detalhado na seção 6.2.2.1. Havendo uma igualdade de triangulação, buscam-se outros padrões semelhantes. Estes padrões são a distância euclidiana entre MA_i e MV_{i+1} ($DistMAMV_2_i$), a direção de MV_{i+1} (θ_{mvi+1}) a quantidade de cristas entre MV_{i+1} e MA_i (rc_{mvi+1}). Se $V_i(DistMAMV_2_i, \theta_{mvi+1}, rc_{mvi+1}) = V_b(DistMAMV_b, \theta_{mab+1}, rc_{mvb+1})$ então um padrão é localizado. Dois acumuladores são incrementados: i) o acumulador de pontos (AcumPont) e ii) o acumulador de triângulos (AcumTriang). Os critérios de término da busca são: i) quando esgotam-se as minúcias em um dos vetores ou ii) quando são encontrados três triângulos, daí o algoritmo garante que os vetores pertencem à mesma impressão digital.

Ao término do processo de busca, é verificado a quantidade de triângulos localizados. Para cada quantidade, são acrescidos "pontuações" de diferentes pesos. Quanto mais triângulos, mais pontos o acumulador AcumPont terá. Assim, o algoritmo retornará a diferença entre uma pontuação limite e o valor de AcumPont.

6.2.2.1 Triangulação

O processo denominado de triangulação, na realidade, é o cálculo dos dois ângulos α e β (figura 6.8) que são formados pelo posicionamento das minúcias $m1$, $m2$ e $m3$ (este agrupamento de minúcias tem formato de um triângulo. Ver figura 6.9 que representam respectivamente MA_i , MV_i e MV_{i+1} ou MA_b , MV_b e MV_{b+1}).

Inicialmente têm-se as minúcias em suas posições originais, como mostrado na figura 6.10. Nela vê-se as três minúcias citadas $m1$, $m2$ e $m3$. Cada uma com sua coordenada original em

relação à coordenada 0,0.

Para o cálculo dos ângulos, é escolhida uma minúcia, suponha $m1$, que terá sua coordenada transladada para 0,0. Imaginando um plano cartesiano, esta minúcia passa a ser a origem deste plano (figura 6.11). As outras duas minúcias, $m2$ e $m3$, ganham novas coordenadas em relação à $m1$. As novas coordenadas são obtidas pela equação

$$x' = x2 - x1 \text{ e } y' = y2 - y1 \quad (6.1)$$

onde x' e y' representam as novas coordenadas de $m2$ e $m3$, $x2$ e $y2$ são as coordenadas originais das minúcias $m2$ e $m3$ e $x1$ e $y1$ é a coordenada original de $m1$. A partir daí é possível calcular os ângulos α e β para $m1$. Entretanto, somente com estes ângulos calculados, não é possível determinar a similaridade das impressões digitais. É necessário também calcular α e β para $m2$ e $m3$. Quando os ângulos são calculados para $m2$, esta passa ser a origem do plano cartesiano e as demais minúcias têm suas coordenadas recalculadas em função a $m2$. Isto também ocorre para $m3$ sendo a origem do plano.

Se tanto a impressão digital de entrada, quanto a do banco de dados tiverem estes ângulos, conclui que existe uma certa similaridade entre elas. Se forem encontrados 3 grupos destes ângulos, o algoritmo retorna que ambas impressões digitais são da mesma pessoa.

6.3 Material

Para avaliar o desempenho do MFIS, foram elaborados três tipos de testes: i) Curvas ROC (*Receiver Operating Characteristic*), ii) Teste de Robustez e iii) Teste de Desempenho.

Os testes foram realizados com duas bases de impressões digitais obtidas do evento *Fingerprint Verification Competition* (FVC). Foram utilizados os bancos DB1 dos anos 2000 e 2002. Os banco de dados possuem 800 impressões digitais que são formadas a partir de 100 classes com

Figura 6.8: Ângulos calculados entre as minúcia alvo e minúcias vizinhas.

Figura 6.9: Conjunto de minúcias em forma de triângulo usado para o cálculo dos ângulos.

8 imagens cada. Nas figuras 6.12 e 6.13, pode-se ver exemplos das imagens da classe 10 dos dois banco de dados.

Para os mesmos usou-se um computador com processador AMD Athlon 2.0, 512 MB de memória RAM. O sistema operacional foi o RedHat Linux 9.0 iniciado no nível de execução 1.

Figura 6.10: Minúcias nas posições originais.

Figura 6.11: Novas coordenadas das minúcias e seus respectivos ângulos

6.4 Resultados

Nesta seção, serão apresentados os resultados dos testes realizados entre a metodologia proposta no MFIS e a proposta por Bozorth.

Figura 6.12: Exemplo das Imagens do Banco de Dados de 2000 do ano FVC

6.4.1 Curvas ROC (Receiver Operating Characteristic)

Para avaliação do MFIS, foi utilizada a metodologia proposta inicialmente no FVC2000(MAIO et al., 2000) que consistiu em calcular a curva ROC do sistema e a taxa EER (*Equal Error Rate*) obtidos através de 2800 comparações denominadas de genuínos (uma imagem da classe é comparada com as demais imagens da mesma classe resultando em $(8 \times 7)/2$ testes por classe) e impostores com 4950 comparações (a primeira imagem da classe é comparada com a primeira imagem das outras classes resultando em $(100 \times 99)/2$ testes). Dos resultados obtidos entre genuínos e impostores, obtém-se os erros FAR (*False Accept Rate*) que indicam os impostores que são aceitos como genuínos e FRR (*False Reject Rate*) que indica os genuínos que são rejeitados como impostores. A curva ROC é traçada por FAR em função de FRR. O EER é resultante do ponto em que $FAR(t) = FRR(t)$, onde t é um threshold que varia de 0 a 1.

O sistema MFIS foi comparado com o software Bozorth3 desenvolvido pelo NIST(GARRIS et

Figura 6.13: Exemplo das Imagens do Banco de Dados do ano 2002 do FVC

al., 2002). A comparação direta foi possível pois ambos utilizam o mesmo sistema de detecção de minúcias que é o software *mindtct* também desenvolvido pelo NIST.

Nas figuras 6.14 e 6.15 tem-se os gráficos, nas escalas *loglog* (6.14), normal (6.15a) e ampliada (6.15b) respectivamente, das curvas ROC do DB1 do FVC do ano 2000. O EER do MFIS foi de 4.9% contra 7.2% do Bozorth3.

Figura 6.14: Curva ROC em escala loglog comparando os resultados entre os sistemas MFIS e o Bozorth3 sobre o banco de dados DB1 FVC 2000

(a)

(b)

Figura 6.15: Curvas ROC em escala a) normal e b) ampliada comparando os resultados entre os sistemas MFIS e Bozorth3 sobre o banco de dados DB1 FVC do ano 2000

As figuras 6.16 e 6.17, mostram as curvas ROC comparando os resultados dos sistemas aplicadas sobre o DB1 do FVC2002. Novamente os gráficos estão nas escalas *loglog* (6.16, normal (6.17a) e ampliada (6.17b)). O EER do sistema MFIS foi de 2.0% contra 2.7% do Bozorth3.

Figura 6.16: Curvas ROC em escala loglog comparando os resultados entre os sistemas MFIS e Bozorth3 sobre o banco de dados DB1 FVC do ano 2002

(a)

(b)

Figura 6.17: Curvas ROC em escala a) normal e b) ampliada comparando os resultados entre os sistemas MFIS e Bozorth3 sobre o banco de dados DB1 FVC 2002

Os resultados obtidos indicam que a taxa de Falsa-Rejeição (FRR) do MFIS possui valor bastante baixo, sendo sua utilização indicada para aplicações forenses no reconhecimento de indivíduos suspeitos, pois nos resultados retornados aumenta-se a probabilidade do possuidor da impressão digital estar entre eles. Devido ao baixo EER, outra utilização é em sistemas comerciais para controle de acesso de pessoas aumentando a confiabilidade do sistema e evitando transtornos de se rejeitar pessoas previamente cadastradas como se fossem impostores.

6.4.2 Robustez de Recuperação das Imagens

A maioria dos algoritmos de reconhecimento de impressões digitais baseados em minúcias partem da premissa que tanto o *template* armazenado no banco de dados quanto a imagem de entrada possuem o mesmo tamanho. Entretanto nem sempre isto é possível. Considerando, por exemplo, o caso de impressões latentes (que são obtidas em cenas de crimes). Elas nem sempre possuem o tamanho total da impressão digital.

Reconhecimento de impressões digitais envolvendo imagens incompletas apresenta vários desafios (JEA; GOVINDARAJU, 2005): i) o número de minúcias é menor, diminuindo a capacidade do sistema; ii) é provável a ausência de pontos singulares (núcleos e deltas), porém nem sempre um algoritmo irá fazer uso disso; iii) ambientes sem controle resulta imagens incompletas sem orientação de posicionamento. Distorções devido à elasticidade e à umidade são introduzidas como características presente na pele humana.

Como no algoritmo proposto para o sistema MFIS não foram empregadas coordenadas da localização das minúcias, mas sim as distâncias entre elas, o mesmo é independente de posicionamento das minúcias. O resultado disso é uma maior robustez na recuperação de imagens incompletas.

Para testar a robustez, foram selecionadas 10 classes, as quais são a 10, 20, 30, 40, 50, 60, 70, 80, 90 e 100. Para cada classe, foi escolhida uma imagem e esta foi dividida em imagens de tamanhos diferentes. Cada novo tamanho representa uma porcentagem da imagem original. As

porcentagens definidas foram 20, 30, 40, 50, 60, 70, 80 e 90 por cento. Na figura 6.18 tem-se um exemplo da divisão de uma imagem.

Figura 6.18: Exemplo de imagens da classe 10 do FVC2002 divididas nas porcentagens de a) 20%, b) 30%, c) 40%, d) 50%, e) 60%, f) 70%, g) 80% e h) 90% da imagem original

Foram realizados dois tipos de testes de robustez. No primeiro, o objetivo foi de determinar com que porcentagem da imagem original o MFIS conseguiria recuperar sua correspondente do banco de dados. A figura 6.19 apresenta dois gráficos de barras com resultados obtidos de porcentagem de acerto sobre os bancos FVC2000 e FVC2002. No eixo horizontal tem-se a porcentagem que foi dividida a imagem. Este eixo indica qual o menor tamanho da imagem é necessário para o MFIS localizar a mesma no banco de dados. No eixo vertical tem-se a porcentagem de acerto de recuperação. Aqui, cada impressão digital recuperada corretamente é computada como acerto. O número máximo de acerto é oito, que representa 100% das imagens

da classe. Tomando como exemplo a figura 6.19(a), vê-se que é necessário pelo menos 40% do tamanho da imagem para que o sistema possa reconhecer a impressão digital. Já na figura 6.19(b), o tamanho mínimo da imagem é de 30%. Isto se deve à melhor qualidade das imagens disponíveis na base do FVC2002.

O segundo tipo de teste foi calcular o EER entre o MFIS e Bozorth. A metodologia adotada para o cálculo do EER é a mesma realizada na seção 6.4.1. Entretanto, existe uma diferença na quantidade de comparações realizadas. O número de genuínos correspondentes a cada tamanho da imagem é obtido realizando-se 80 comparações (8 testes por classe) e o número de impostores com 495 comparações $((10 \times 99)/2)$.

Figura 6.19: Porcentagem de acerto do MFIS dos banco de dados FVC2000 e FVC2002.

Na tabela 6.1 têm-se os resultados do EER dos softwares MFIS e Bozorth para cada porcentagem de tamanho. As tabelas indicam que quanto maior é o tamanho da imagem de entrada, menor é o EER. O MFIS em todos os casos possui um EER inferior ao Bozorth.

Com esses resultados, novamente tem-se o uso do MFIS apropriado para aplicações forenses, principalmente às voltadas para cenas de crimes, onde as imagens das impressões recolhidas, nem sempre estão inteiras. Esta afirmação pode ser confirmada pelos gráficos da figura 6.19, pois no gráfico 6.19(a), uma impressão digital pode ser reconhecida com 40% do seu tamanho original e no gráfico da 6.19(b), com apenas 30% de seu tamanho.

Tabela 6.1: Taxa EER entre os softwares MFIS e o Bozorth

	40%	50%	60%	70%	80%	90%
MFIS	20	20	18	14,6	10	10
Bozorth	35,5	29,8	24,1	15,9	15,3	12,8

(a) FVC2000

	30%	40%	50%	60%	70%	80%	90%
MFIS	20	7,8	6,9	6,4	5,0	2,2	2,2
Bozorth	40,5	30	19	18,7	16,4	16,1	15,9

(a) FVC2002

6.4.3 Teste de desempenho

O objetivo do teste é analisar o desempenho dos sistemas comparados no sentido de se saber qual deles leva menos tempo para recuperar na base de dados uma impressão digital de entrada.

Como citado na seção 6.3, usou-se o sistema operacional RedHat Linux 9.0 iniciado no nível de execução 1. Neste nível de execução, com exceção dos processos de sistemas operacionais, não existe processos sendo executados. Isso evita problemas de concorrência entre processos. O que poderia influenciar no resultado dos testes. Outro ponto importante a ser considerado é o tempo gasto na busca de dados armazenados no disco rígido que é muito superior ao tempo de busca de dados já na memória RAM. O Linux faz o uso de cache de dados. Uma vez que os dados são levados do disco rígido para memória RAM, estes ficam nela até a necessidade do uso do espaço da memória pra outros dados. O que não aconteceu neste caso.

A metodologia empregada para realização dos testes foi a seguinte: foram selecionadas a primeira imagem das classes 10, 20, 30, 40, 50, 60, 70, 80, 90 e 100 dos banco de dados FVC2000 e FVC2002. Para cada imagem foram recuperadas as quantidades 1, 2, 3, 4, 5, 6, 7, 8 e 800 imagens de impressões digitais. Estas quantidades foram escolhidas, pois, para cada classe, existem 8 imagens pertencente a ela e na busca por 800 imagens o acesso é sequencial. Para cada quantidade foram feitas 15 buscas na base. Uma vez que o software MFIS usa uma árvore

de indexação, ele tem vantagens sobre o Bozorth no sentido de recuperar menor quantidade de imagens. Realizadas as buscas, calcula-se uma média para àquela busca. Vale ressaltar que antes dos testes contabilizados, são feitas 5 buscas para cada classe e quantidade que não são contabilizadas. O objetivo é para garantir que os dados se encontrem na memória RAM em forma de cache. Isto evita que os tempos contabilizados não sofrerão impactos negativos pela busca no disco rígido.

As figuras e 6.21 mostram alguns dos resultados obtidos da comparação dos testes de desempenho entre os softwares MFIS e Bozorth no banco de dados do FVC2000. Os gráficos indicam o tempo gasto em segundos pela quantidade de impressões digitais procuradas. Uma vez que o software Bozorth não é indexado, seu tempo de busca sempre será linear, independente da quantidade de impressões digitais procuradas. Já o MFIS, que opera sobre uma base de dados indexada, obtêm vantagens na busca de poucas impressões digitais. Entretanto, quando há necessidade de buscar todas as impressões digitais, o tempo gasto é maior que o do Bozorth. Isto se deve ao fato que para percorrer todo o índice calculando a distância entre as impressões digitais, é mais demorado que percorrer todas as impressões digitais de forma sequencial.

Figura 6.20: Resultados dos testes de desempenho das classes 10 e 30 da base de dados DB1 do FVC2000

Figura 6.21: Resultados dos testes de desempenho das classes 50, 70 e 90 da base de dados DB1 do FVC2000

Nos gráficos das figuras 6.22 e 6.23, têm-se os resultados para o banco de dados do FVC2002. Novamente o MFIS obteve vantagens de desempenho na recuperação de um número pequeno de impressões digitais. Note-se que aqui os tempos do MFIS apresentam, em algumas situações, vantagens mesmo na recuperação de toda a base de dados (figuras 6.22a e 6.23a).

(a) Classe 10

(b) Classe 30

Figura 6.22: Resultados dos testes de desempenho das classes 10 e 30 da base de dados DB1 do FVC2002

(a) Classe 50

(b) Classe 70

(c) Classe 90

Figura 6.23: Resultados dos testes de desempenho das classes 50, 70 e 90 da base de dados DB1 do FVC2002

Figura 6.24: Média dos testes de desempenho de todas as classes das bases de dados DB1 do FVC2000 e FVC2002

Na figura 6.24, observa-se a média de todos os tempos gastos para recuperação das 10 classes dos bancos de dados DB1 do FVC2000 e FVC2002. É bem perceptível a diferença dos tempos entre os sistemas com relação à base de dados utilizada. No gráfico 6.24(a), o MFIS só obteve vantagens significativas recuperando até 6 impressões similares, depois há um empate nos tempo e a partir daí, o Bozorth leva vantagens. Já no gráfico da figura 6.24(b), a vantagem do MFIS é bem acentuada. Ele é superior na recuperação das 8 impressões digitais da mesma classe. Uma explicação para esta diferença de desempenho é a qualidade das imagens de ambos os bancos de dados. Uma vez que no FVC2002 as imagens tem qualidade superior ao FVC2000, o extrator de características obtém melhores resultados impactando positivamente o processo de busca do MFIS, no sentido que as classes são melhores segmentadas. Conclui-se que é vantajoso a utilização de uma árvore de busca no processo de indexação de impressões digitais em relação à realização de uma busca seqüencial em toda a base.

6.5 Considerações Finais

Neste capítulo foi apresentada a metodologia proposta nesta tese que consistiu em desenvolver um algoritmo original de reconhecimento de impressões digitais que opera em espaço métrico e utilizá-lo dentro da árvore de busca métrica *Slim-Tree* para o reconhecimento e indexação de impressões digitais.

Foram abordadas três classes de problemas apresentadas em pesquisas recentes com impressões digitais: i) o reconhecimento, ii) robustez no processo de recuperação com parte da imagem original e iii) indexação.

Para comprovar a eficiência do algoritmo desenvolvido, o mesmo foi comparado com o proposto por Bozorth. Os testes foram realizados com imagens dos Banco de Dados 1 disponibilizadas no evento *Fingerprint Verification Competition* (FVC) dos anos de 2000 e 2002.

Os desempenhos de ambos os sistemas foram avaliados pelos resultados obtidos com as curvas ROC e o valor do *Equal Error Rate* (EER). Para o banco de dados do FVC2000, o EER foi de 4.9% para o algoritmo proposto contra 7.2% do Bozorth. No banco de dados do FVC2002, o EER para o algoritmo proposto foi de 2% contra 2,7% no Bozorth. Analisando-se as curvas ROC conclui-se que o sistema proposto pode ser indicado com vantagens para aplicações forenses de identificação de impressões digitais ou em sistemas comerciais de controle de acesso de pessoas devido sua baixa taxa de Falsa-Rejeição (FRR). Outra informação que reforça seu uso em aplicações forenses são os resultados dos testes de robustez. Os mesmos indicam que o sistema foi capaz de reconhecer uma impressão digital utilizando-se apenas 30% da imagem original. Com a indexação da base de dados constatou-se uma redução no tempo de busca, comparativamente ao acesso seqüencial. O tempo gasto do sistema proposto foi menor que o Bozorth no que diz respeito a recuperar pequenas quantidades de impressões da mesma classe.

Capítulo 7

Conclusões e Trabalhos Futuros

O processo automatizado de reconhecimento de pessoas, normalmente, é baseado em senhas e objetos. As desvantagens desses métodos são inúmeras: podem ser esquecidos, roubados, perdidos, falsificados, etc. Uma outra maneira mais segura de reconhecimento de pessoas é a biometria, ou seja, utilizar características presentes no indivíduo para assegurar sua autenticidade.

Dentre as possíveis características biométricas (impressão digital, retina, íris, mãos, voz, etc), a impressão digital é a que tem recebido grande interesse nas pesquisas. Isso deve-se ao fato de ser única e imutável, ser de fácil aceitação, não intrusiva, entre outras. Com o crescente uso das impressões digitais em sistemas biométricos, cresceu também o volume de dados armazenados referentes a elas. Para agilizar as buscas nas bases de dados, pesquisas envolvendo indexação de impressões digitais tem tido atenção especial. A idéia é que somente impressões digitais similares sejam comparadas, o que diminuiria o tempo de busca.

Este trabalho teve como proposta o desenvolvimento de um algoritmo inovador para o reconhecimento de impressões digitais com indexação das mesmas em base de dados. O algoritmo proposto foi projetado para operar em espaço métrico. Com esta característica, ele pode ser utilizado na árvore de busca métrica Slim-tree. Assim, o algoritmo proposto além de reconhecer,

permite a indexação das impressões digitais.

Para comprovação de sua eficiência, foram realizados testes com imagens de impressões digitais disponibilizadas no evento *Fingerprint Verification Competition* (FVC) dos anos de 2000 e 2002 e os resultados obtidos foram comparados com o algoritmo proposto por Bozorth.

A avaliação dos sistemas se deu pelos resultados obtidos pelas curvas ROC (*Receiver Operating Characteristic*) e o EER (*Equal Error Rate*), sendo que este foi de 4.9% para o algoritmo proposto contra 7.2% do Bozorth sobre o banco de dados do ano de 2000. Para o banco de dados do ano de 2002, o algoritmo proposto obteve 2% contra 2,7% do Bozorth. Em relação as curvas obtidas, conclui-se que o algoritmo proposto é indicado para aplicações forense de identificação de impressões digitais ou em sistemas comerciais de controle de acesso de pessoas por apresentar uma baixa taxa de Falsa-Rejeição (FRR). Tratando ainda de aplicações forenses, os resultados dos testes de robustez demonstram que com apenas 30% da área de uma imagem da impressão digital, foi possível reconhecê-la. Nos testes realizados com a base indexada, constatou-se uma redução no tempo de busca comparados ao acesso seqüencial. O tempo gasto do sistema proposto foi menor que o Bozorth no que diz respeito a recuperar pequenas quantidades de impressões digitais pertencentes a uma mesma classe.

7.1 Contribuições da Tese

Este trabalho apresentou as seguintes contribuições importantes para a área de biometria:

1. Desenvolvimento de um algoritmo original de reconhecimento de impressões digitais baseado em espaço métrico possibilitando sua utilização em uma árvore de busca métrica para indexação de impressões digitais reduzindo os tempos de buscas de impressões digitais, fato que, está se tornando necessário, devido o aumento no volume das informações biométricas provenientes de impressões digitais.

2. Desenvolvimento de um algoritmo robusto que mesmo com 30% ou 40% da imagem original, apresenta resultados de reconhecimentos satisfatórios.
3. Redução do EER (*Equal Error Rate*) relativamente a uma metodologia amplamente empregada (Bozorth) o que o qualifica para aplicações comerciais de controle de acesso.
4. Redução da FRR (*False Reject Rate*) o que o qualifica como excelente para aplicações forenses.

7.2 Sugestões de Trabalhos Futuros

As sugestões para trabalhos futuros são:

- Utilizar o algoritmo desenvolvido em outras árvores de buscas métricas, comparando os tempos envolvidos.
- Pesquisar a robustez em relação ao de rotacionamento das impressões digitais.
- Utilizar outras bases de imagens.
- Utilizar bases oriundas de sensores com tecnologias diferentes.

Referências Bibliográficas

BALLAN, M.; SAKARYA, F. A. A fingerprint classification technique using directional images. *University of Texas at Austin, Department of ECE*, 1998.

BAYER, R.; MCCREIGHT, E. M. Organization and maintenance of large ordered indexes. *Acta Informatic*, v. 1, n. 3, p. 173–189, 1972.

BHANU, B.; TAN, X. Fingerprint indexing based on novel feature of minutiae triplets. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, v. 25, n. 5, p. 616–622, may, 2003.

BOZKAYA, T.; OZSOYOGLU, M. Distance-based indexing for high-dimensional metric spaces. *Proceedings of the 1997 ACM International Conference on Management of Data (SIGMOD)*, p. 357–368, 1997.

BRAGHIN, C. Biometric authentication. *University of Helsinki, Department of Computer Science*, 2001. Disponível em: <citeseer.ist.psu.edu/436492.html>.

BRAHMI, D.; ZIOU, D. Improving CBIR systems by integrating semantic features. *Proceedings of the First Canadian Conference on Computer and Robot Vision*, 2004.

BUKHARD, W.; KELLER, R. Some approaches to best-match file searching. *Comm. of the ACM*, v. 16, n. 4, p. 230–236, 1973.

CHAVEZ, E. et al. Searching in metric space. *ACM Computing Surveys*, v. 33, n. 3, p. 273–321, 2001.

CHIKKERUR, S.; CARTWRIGHT, A. N.; GOVINDARAJU, V. K-plet and cbfs: A graph based fingerprint representation and matching algorithm. *Center for Unified Biometrics and Sensors, University at Buffalo, NY, USA*, 2005.

CIACCIA, P.; PATELLA, M.; ZEZULA, P. M-tree: An efficient access method for similarity search in metric spaces. *Proceedings of 23rd International Conference on Very Large Data Bases (VLDB)*, p. 426–435, 1997.

FINKEL, R. A.; BENTLEY, J. L. Quad Trees: A data structure for retrieval on composite keys. *Acta Informatica*, v. 8, p. 1–9, 1974.

GARRIS, M. et al. User's guide to nist fingerprint image software (NFIS). *National Institute of Standards and Technology*, 2001. Disponível em: <fingerprint.nist.gov>.

GARRIS, M. et al. User's guide to nist fingerprint image software (NFIS). *National Institute of Standards and Technology*, 2002. Disponível em: <fingerprint.nist.gov>.

GERMAIN, R. S.; CALIFANO, A.; COLVILLE, S. Fingerprint matching using transformation parameter clustering. *IEEE Computational Science and Engineering*, v. 4, n. 4, p. 42–49, 1997.

GUDEVADA, V.; RAGHAVAN, V. Content-based image retrieval systems. *IEEE Computer*, p. 18–22, September, 1995.

GUTTMAN, A. R-trees: a dynamic index structure for spatial searching. *Proceedings of the ACM SIGMOD Conference*, p. 47–57, Boston, June, 1984.

HAYKIN, S. Neural networks, a comprehensive foundation. *Prentice Hall International, Inc., Upper Saddle River, NJ*, 1999.

HONG, L.; WAN, Y.; JAIN, A. K. Fingerprint image enhancement: Algorithm and performance evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 20, n. 8, p. 777–789, 1998.

JAIN, A.; HONG, L. On-line fingerprint verification. *Proceedings of the 13th International Conference on Pattern Recognition*, v. 3, p. 596–600, 1996.

JAIN, A. et al. An identity authentication system using fingerprints. *Proceedings of the IEEE*, v. 85, n. 9, p. 1365–1388, september, 1997.

JAIN, A.; PANKANTI, S. Fingerprint classification and matching. *Michigan State University*, 2000.

JAIN, A. et al. Filterbank-based fingerprint matching. *IEEE Transactions on Image Processing*, v. 9, n. 5, p. 846–859 May, 2000.

JAIN, A. K.; PRABHAKAR, S.; HONG, L. A multichannel approach to fingerprint classification. *IEEE Trans. Pattern Anal. Machine Intell.*, v. 21, n. 4, p. 348–359, 1999.

JEA, T.; GOVINDARAJU, V. Partial fingerprint recognition based on localized features and matching. *University at Buffalo, the State University of New York*, 2005.

KAWAGOE, M.; TOJO, A. Fingerprint pattern classification. *Pattern Recognition*, v. 17, n. 3, p. 295–303, 1984.

KRUSKAL, J. B. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proc. American Math Soc.*, v. 7, p. 48–50, 1956.

LI, D.; SIMSKE, S. Shape retrieval based on distance ratio distribution. *HP Laboratory Palo Alto*, 2002.

LIU, S.; SILVERMAN, M. A practical guide to biometric security technology. *IEEE Computer*, January-February, 2001.

LIU, Y. et al. Region-based image retrieval with high-level semantic color names. *Proceedings of the 11th International Multimedia Modelling Conference*, 2005.

MAIO, D.; MALTONI, D. A structural approach to fingerprint classification. *IEEE Proceedings of ICPR'96*, p. 578–585, 1996.

MAIO, D. et al. Fvc2000: Fingerprint verification competition. *Biolab internal report, University of Bologna, Italy*, p. September, 2000.

MATYAS Jr., V.; RIHA, Z. Biometric authentication systems. *Universitas Masarykiana*, 2000.

NAVARRO, G. Searching in metric spaces by spatial approximation. *The Very Large Database Journal*, n. 11, p. 28–46, 2002.

NEGIN, M.; CAMUS, T. An iris biometric system for public and personal use. *IEEE computer*, special issue, February, 2000.

OTTERMAN, M. Approximate matching with high dimensionality r-trees. *M.Sc Scholarly paper, Dept. of Computer Science, Univ. of Maryland, Collage Park, MD. Supervised by C. Faloutsos*, 1992.

PETRAKIS, E.; FALOUTSOS, C. *Similarity Searching In Large Image Database*. 1995.

PHILLIPS, P. J. et al. An introduction to evaluating biometric systems. *IEEE Computer*, p. 56–63 February, 2000.

RATHA, N.; CHEN, S.; JAIN, A. K. Adaptive flow orientation based feature extraction in fingerprint images. *Pattern Recognition*, v. 28, n. 11, p. 1657–1672, November, 1995.

ROBINSON, J. T. A search structure for large multidimensional dynamic indexes. *ACM International Conference on Management of Data (SIGMOD)*, p. 10–18, 1981.

ROSS, A.; JAIN, A.; QIAN, Z. Information fusion in biometrics. *Lecture Notes in Computer Science*, v. 24, n. 13, p. 2115–2125, 2003.

ROSS, A.; JAIN, A.; REISMAN, J. A hybrid fingerprint matcher. *Pattern Recognition*, v. 36, p. 1661–1673, 2003.

SAHA, S. K.; DAS, A. K.; CHANDA, B. Cbir using perception based texture and colour measures. *Proceedings of the 17th International Conference on Pattern Recognition*, 2004.

SANTOS FILHO, R.; TRAINA, A.; TRAINA Jr. C.; FALOUTSOS C. Similarity search without tears: The OMNI family of all-purpose access methods. *International Conference on Data Engineering ICDE. IEEE Computer Society*, p. 623–630, 2001.

SMEULDERS, A. et al. Content-based image retrieval at the end of the early years. *Transactions on Pattern Analysis and Machine Intelligence*, v. 22, n. 12, p. 1349–1380, December, 2000.

TISSE, C. et al. *Person Identification Technique Using Human Iris Recognition*. 2002. Disponível em: <citeseer.ist.psu.edu/tisse02person.html>.

TRAINA Jr., C.; TRAINA, A.; SEEGER, B.; FALOUTSOS C. Slim-Trees: High performance metric trees minimizing overlap between nodes. *Lecture Notes in Computer Science*, v. 1777, p. 51–65, 2000. Disponível em: <citeseer.ist.psu.edu/traina00slimtrees.html>.

UHLMANN, J. Satisfying general proximity/similarity queries with metric trees. *Information Processing Letter*, v. 40, n. 4, p. 175–179, 1991.

VAUGHAN-NICHOLS, S. J. Voice authentication speaks to the marketplace. *IEEE Computer*, March, 2004.

WACKTLAR, H. D. et al. Intelligent access to digital video:informedia project. *IEEE Computer*, v. 29, n. 3, p. 46–52, 1996.

WANG, C.; GAVRILOVA, M. L. Delaunay triangulation algorithm for fingerprint matching. *3rd International Symposium on Voronoi Diagrams in Science and Engineering (ISVD'06)*, p. 208–216, July, 2006.

YIANILOS, P. N. Data structures and algorithms for nearest neighbor search in general metric spaces. *ACM Symposium on Discrete Algorithms (SODA)*, p. 311–321, 1993.

ZACHARY, J.; IYENGAR, S.; BARTHEN, J. Content based image retrieval and information theory: A general approach. *Journal of the American Society for Information Science and Technology*, v. 52, n. 10, p. 840–852, 2001.

ZHANG, D. Improving image retrieval performance by using both color and texture features. *Proceedings of the Third International Conference on Image and Graphics*, 2004.

ZHAO, F.; TANG, X. Preprocessing for skeleton-based fingerprint minutiae extration. *CISST Internation Conference*, p. 742–745, 2002.

ZHAO, W. et al. *Face recognition: A literature survey*. 2000. Disponível em: <citeseer.ist.psu.edu/zhao00face.html>.