

FERNANDA BAUMGARTEN RIBEIRO DO VAL
PRISCILA RIBEIRO MARCELINO

**SISTEMA ADAPTATIVO DE RECONHECIMENTO BIOMÉTRICO POR
IMPRESSÃO DIGITAL**

São Paulo
2014

FERNANDA BAUMGARTEN RIBEIRO DO VAL
PRISCILA RIBEIRO MARCELINO

**SISTEMA ADAPTATIVO DE RECONHECIMENTO BIOMÉTRICO POR
IMPRESSÃO DIGITAL**

Trabalho de formatura apresentado à
Escola Politécnica da Universidade
de São Paulo para obtenção do título
de Engenheiro Eletricista

São Paulo
2014

FERNANDA BAUMGARTEN RIBEIRO DO VAL

PRISCILA RIBEIRO MARCELINO

SISTEMA ADAPTATIVO DE RECONHECIMENTO BIOMÉTRICO POR IMPRESSÃO DIGITAL

Trabalho de formatura apresentado à
Escola Politécnica da Universidade
de São Paulo para obtenção do título
de Engenheiro Eletricista

Área de concentração:
Engenharia de computação e
sistemas digitais

Orientador:
Prof. Dr. João José Neto

São Paulo

2014

Agradecimentos

Nossos agradecimentos ao professor orientador e amigo, João José Neto, pela atenção e suporte durante todo o projeto, além de sempre nos incentivar a dar o nosso melhor e nos motivar lembrando do potencial deste projeto. A sua contribuição com ideias, sugestões e *feedbacks* foi muito importante para a qualidade do trabalho que realizamos, não somente neste projeto.

Agradecemos também ao Rafael Camargo Leite, estudante da Escola Politécnica, pelo suporte e apoio durante todo o projeto, em especial pela sua contribuição no esclarecimento de diversas dúvidas técnicas relacionadas ao ambiente de desenvolvimento do software. A sua participação e seu companheirismo foram cruciais para o desenvolvimento deste projeto.

Queremos agradecer também à contribuição que Albert Nissimoff, diretor da empresa Control ID, fez ao projeto por meio de explicações e esclarecimentos técnicos relacionados aos algoritmos tradicionais de identificação por impressão digital. Sua disponibilidade e interesse no projeto, combinados a sua vasta experiência na área, auxiliaram nos resultados alcançados.

Por fim, não poderíamos deixar de agradecer às nossas famílias e amigos, por todo apoio e incentivo durante todos esses anos de Escola Politécnica. Sem dúvida, ter vocês ao nosso lado, tornou essa árdua jornada mais feliz.

Abstract

Identification of individuals, with practicality and reliability, is an important need of our society, especially when it comes to provide access to restricted places and systems. Based on this, there are increasing studies about biometric authentication, which usually combines easiness for users and security. This monograph focus on proposing the application of adaptive concepts on traditional algorithms of fingerprint authentication, the most used biometric technique, in order to improve the performance of these algorithms through the reduction of false decisions. The results were positive, showing the reduction of false acceptances or false rejections, according to which metric has the most priority for each context.

Resumo

A identificação de indivíduos, com praticidade e confiabilidade, é uma necessidade importante da nossa sociedade, especialmente quando se trata de prover acesso a locais e sistemas restritos. Com base nisso, há um número crescente de estudos sobre autenticação biométrica, que em geral combina segurança e facilidade para os usuários. Este trabalho é focado em propor a aplicação de conceitos de adaptatividade em algoritmos tradicionais de reconhecimento de impressão de digital, a técnica biométrica mais comum, com o objetivo de melhorar o desempenho desses algoritmos por meio da redução de decisões errôneas.

Os resultados obtidos com o projeto foram positivos, visto que se observa redução de falsas aceitações ou falsas rejeições, de acordo com qual a métrica mais prioritária para cada contexto de aplicação.

Sumário

Sumário.....	4
Lista de Figuras.....	7
1 Introdução.....	10
1.1 Motivação	10
1.2 Objetivo	11
1.3 Organização	12
2 Metodologia	14
2.1 Pesquisa	14
2.2 Levantamento de Demandas de Mercado	15
2.3 Desenvolvimento	15
2.3.1 Organização do desenvolvimento do projeto	16
2.3.2 Cronograma	17
3 Conceitos e tecnologias.....	18
3.1 Sensores de captura de impressões digitais	18
3.1.1 Ultrassônico	18
3.1.2 Ótico	18
3.1.3 Sensores de Estado Sólido	19
3.2 BDs de imagens de digitais	21
3.3 Classificação de Digitais	22
3.4 Reconhecimento de impressões digitais por minúcias	23
3.5 Reconhecimento de Padrões	25
3.6 Adaptatividade	27
4 Especificação do Sistema.....	29
4.1 Métricas de desempenho	29
4.2 Sub-sistema 1: Sem pré-processamento e não adaptativo	31
4.3 Sub-sistema 2: Com pré-processamento e não adaptativo	32
4.4 Sub-sistema 3: Com pré-processamento e adaptativo	33
4.5 Escopo	35
4.6 Requisitos Funcionais	36
4.7 Requisitos Não-Funcionais	37
4.8 Estrutura do Banco de Dados	37
4.9 Imagem de entrada	39

4.10	Pré-processamento	40
4.10.1	Adição de borda	42
4.10.2	Janelamento	42
4.10.3	Normalização convencional (não-adaptativa)	44
4.10.4	Normalização adaptativa	45
4.10.5	Binarização	46
4.10.6	Mapa de orientação	46
4.10.7	Mapa de frequência	48
4.10.8	Filtro de Gabor	51
4.10.9	Afinamento	52
4.11	Extração de minúcias	53
4.11.1	Identificação de candidatas a minúcias	54
4.11.2	Eliminação de falsas minúcias	56
4.12	<i>Matching</i>	60
4.12.1	Geração da nota de similaridade	60
4.12.2	Decisão com base em nota de corte fixa	61
4.12.3	Decisão com base em nota de corte adaptativa	62
4.13	Interface Gráfica	64
4.13.1	Telas de passo-a-passo do sistema	66
4.13.2	Tela de métricas	69
5	Implementação	71
5.1	C++	71
5.2	Eclipse	71
5.3	QML	72
5.4	Qt Creator	73
5.4.1	Editor de código avançado	73
5.4.2	Depurador visual	74
5.4.3	Construtores de GUI	74
5.5	SQLite	74
5.6	OpenCV	75
5.7	<i>NIST Biometric Image Software (NBIS)</i>	76
6	Testes e Resultados	78
6.1	Metodologia de testes da FVC	78
6.2	Metodologia de testes adotada	80
6.3	Parâmetros adotados	80

6.3.1	Resultados dos testes.....	82
6.3.2	Conclusões sobre os resultados	83
7	Considerações finais.....	85
7.1	Possíveis trabalhos complementares relacionados a reconhecimento de padrões.....	85
7.2	Possíveis trabalhos complementares relacionados à adaptatividade ..	86
7.3	Conclusão	87
8	Referências.....	88

Lista de Figuras

Figura 1 - (a) Bifurcação, (b) Terminação	11
Figura 2 - Sensor Ótico	19
Figura 3 - Sensor Capacitivo	20
Figura 4 - Sensor térmico	20
Figura 5 - Sensor de campo elétrico	21
Figura 6 - Classificação de Tipos de Digitais.....	22
Figura 7 - Minúcias.....	23
Figura 8 - Mapa de Minúcias	24
Figura 9 - Árvore de Minúcias	25
Figura 10 - Eliminação de Falsas Minúcias	25
Figura 11- Diagrama do processo de reconhecimento de padrão.....	26
Figura 12 - Tecnologia Adaptativa.....	28
Figura 13 – Relação entre as principais métricas de desempenho (FAR, EER e FRR)	30
Figura 14 - Operação típica para diferentes aplicações	30
Figura 15- Fluxo de execução do Sub-sistema 1	32
Figura 16 - Fluxo de execução do Sub-sistema 2	33
Figura 17 - Fluxo de execução do Sub-sistema 3	35
Figura 18 - Banco de Dados (versão inicial)	38
Figura 19 - Banco de Dados (versão final)	39
Figura 20 - Diagrama das etapas de pré-processamento da imagem de entrada dos sub-sistemas 2 e 3. (a) Sub-sistema 2 - com pré-processamento de imagem, mas sem adaptatividade -, (b) Sub-sistema 3 - com pré-processamento de imagem e com adaptatividade.	41
Figura 21 – Imagem original (sem bordas brancas), à esquerda; e com a adição de bordas brancas, à direita.	42
Figura 22 – Janelas.....	43
Figura 23 - Estrutura de uma "window"	43
Figura 24 - (a) Imagem original, (b) normalização aplicação de forma convencional, (c) normalização aplicada adaptativamente (por janela)	45
Figura 25 - Binarização	46
Figura 26 - Máscaras de Sobel 3x3 e 5x5, (a) e (c) para obtenção do gradiente horizontal, (b) e (d) para obtenção do gradiente vertical	47
Figura 27 – Modelagem de cristas e vales como um onda de formato senoidal (ALONSO-FERNANDEZ et al., 2003)	49
Figura 28 - DFT aplicada a imagem com 4 ciclos	50
Figura 29 - DFT aplicada a imagem com 16 ciclos	50
Figura 30 - DFT aplicada a imagem complexa.....	51
Figura 31 - Filtro de Gabor	52
Figura 32 - Afinamento.....	53
Figura 33 - Máscaras com padrões de minúcias do MINDTCT.....	55
Figura 34 - Orientação de minúcias	56

Figura 35 - Falsas minúcias. (a) cristas quebradas, (b) ponte, (c) crista curta, (d) crista curta, (e) crista curta, (f) buraco.....	56
Figura 36 - Geração de falsas minúcias no afinamento	57
Figura 37 - Validação de um candidato a terminação. Passo-a-passo do algoritmo, com $W = 11$	58
Figura 38 - Validação de um candidato a bifurcação. Passo-a-passo do algoritmo, com $W = 11$	59
Figura 39 - Cancelamento de falsa minúcia. (a) buraco, (b) spike, com $W = 11$	59
Figura 40 - Distância e orientação relativas entre minúcias	61
Figura 41 - Tradeoff entre métricas de desempenho	64
Figura 42 - Interface de passo-a-passo do sistema: Etapa de Janelamento.....	67
Figura 43 - Interface de passo-a-passo do sistema: Etapa de Normalização	67
Figura 44 - Interface de passo-a-passo do sistema: Etapa de Filtro de Gabor	68
Figura 45 - Interface de passo-a-passo do sistema: Etapa de Binarização	68
Figura 46 - Interface de passo-a-passo do sistema: Etapa de Extração de Minúcias	69
Figura 47 - Interface de passo-a-passo do sistema: Etapa de Matching.....	69
Figura 48 - Interface de métricas dos sistemas.....	70

Lista de Abreviaturas

BD: Banco de Dados

DFT: Discret Fourier Transform

EER: Equal Error Rate

EPUSP: Escola Politécnica da USP

FAR: False Acceptance Rate

FMR: False Matching Rate

FNMR: False Non Matching Rate

FRR: False Rejection Rate

FVC: Fingerprint Verification Competition

GUI: Graphical User Interface

HW: hardware

IDE: Integrated Development Environment

IEEE: Institute of Electrical and Electronics Engineers

LTA: Laboratório de Linguagens e Técnicas Adaptativas

NBIS: NIST Biometric Image Software

NIST: National Institute of Standards and Technology

OPENCV: Open Source Computer Vision Library

PCS: Departamento de Engenharia da Computação e Sistemas Digitais

1 Introdução

Este projeto consiste no desenvolvimento de um sistema de autenticação biométrica por meio de impressão digital, que utiliza técnicas adaptativas, a fim de melhorar seu desempenho.

1.1 Motivação

Atualmente, características físicas do ser humano têm sido utilizadas para identificação de indivíduos. Alguns exemplos são o reconhecimento por meio de íris e de impressão digital. Segundo Castro, a utilização destas características humanas, em comparação aos métodos não biométricos tradicionais, pode aumentar a segurança dos sistemas. Além disso, sistemas biométricos trazem grande praticidade, já que não utilizam senhas, cartões ou outros instrumentos, que podem ser clonados ou roubados facilmente, ou ainda esquecidos.

O reconhecimento por impressão digital é a modalidade mais amplamente utilizada atualmente, não só devido ao seu custo inferior em relação aos demais sistemas biométricos, mas também pela facilidade para o usuário e pela confiabilidade do reconhecimento por digitais.

As impressões digitais de cada indivíduo apresentam características denominadas minúcias, que para cada indivíduo se dispõem em arranjos únicos, que permitem a identificação de um indivíduo em uma população. As linhas que formam o relevo com o desenho da impressão digital são denominadas cristas e as linhas mais profundas são denominadas vales. Algumas das principais e mais comuns minúcias utilizadas atualmente são: (1) quando a crista termina e se inicia um vale – terminação; (2) quando a crista se divide em outras duas – bifurcação. A imagem a seguir ilustra as minúcias descritas.

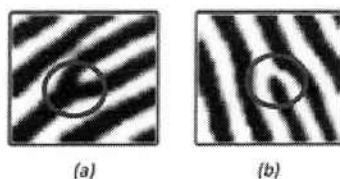


Figura 1 - (a) Bifurcação, (b) Terminação

Fonte: Autor

A qualidade da impressão digital varia entre as pessoas, muitas vezes estando associada à função laboral da mesma. No geral, indivíduos que trabalham realizando esforços com as mãos – como pedreiros, costureiras, operadores de caixa, entre outros – apresentam digitais em que as minúcias têm qualidade inferior às de um indivíduo que não realiza grandes esforços manuais.

Além disso, a digital de cada ser humano varia com o tempo, seja por envelhecimento ou pela aquisição de feridas e cicatrizes. Desta forma, com o passar do tempo os sistemas de identificação passam a rejeitar com maior frequência usuários cadastrados, visto que a impressão digital desses usuários se alterou.

Também é comum o sistema detectar falsas minúcias devido a impurezas no leitor de impressão digital, devido a resíduos no dedo do indivíduos ou mesmo distorção elástica da pele ao ser pressionada contra o sensor. A detecção de falsas minúcias, ou seja, a identificação de terminações e bifurcações que não correspondem ao desenho da digital, dificulta a correta identificação dos usuários.

Para minimizar as decisões de aceitação ou rejeição incorretas, este projeto aplica técnicas adaptativas que melhoram o desempenho do sistema, reduzindo as falsas aceitações ou falsas rejeições.

1.2 Objetivo

O algoritmo de reconhecimento biométrico por impressão digital é feito em três etapas: (a) Pré-processamento para melhorar a qualidade da imagem; (b) Extração

de parâmetros de características individuais; (c) Comparação com dados armazenados e decisão de aceitação ou rejeição do usuário (etapa conhecida como *matching*).

Este projeto tem como objetivo desenvolver e avaliar comparativamente três sub-sistemas de autenticação biométrica por impressão digital: (1) sem pré-processamento de imagem; (2) com as três etapas, implementadas convencionalmente; (3) idem, incluindo técnicas adaptativas para melhorar a qualidade dos resultados.

As técnicas adaptativas permitem que as regras de decisão sejam alteradas em função de erros e acertos do sistema, que são aferidos a partir do *feedback* do operador.

O nível de exigência de cada aplicação pode variar, algumas buscando maior segurança e outras, maior usabilidade. Tendo em vista isso, o Sub-sistema 3, que faz uso de técnicas adaptativas, apresenta três modos de operação: rigoroso, tolerante ou padrão, e ajusta suas regras de forma a minimizar falsas aceitações (no caso do rigoroso) ou falsas rejeições (no caso do tolerante). Não há alteração de regras no caso do modo padrão.

Com a comparação entre os sub-sistemas, é possível analisar o potencial de melhoria de desempenho que adaptatividade pode trazer para a área de reconhecimento biométrico.

Além disso, é feita uma comparação com *benchmarkings*, que são os algoritmos vencedores da competição internacional *Fingerprint Verification Competition (FVC)*.

1.3 Organização

Este documento contém a descrição dos algoritmos referentes aos três sub-sistemas de autenticação biométrica por impressão digital desenvolvidos, incluindo a

explicação das técnicas de tratamento de imagem, extração de minúcias e decisão de aceitação ou rejeição.

Inicialmente são descritas as fases de análise e planejamento do projeto, como a metodologia de pesquisa e desenvolvimento do projeto, os conceitos e tecnologias estudados e a especificação do projeto – escopo, requisito funcionais e não-funcionais, detalhamento das etapas que compõem o sistema (pré-processamento de imagem, extração de minúcias e *matching*) e as tecnologias escolhidas para banco de dados e interface gráfica.

Posteriormente, é detalhada a implementação do projeto, assim como as modificações e adaptações requeridas. Após a implementação do projeto, foram realizados testes a fim de avaliar o desempenho de cada um dos sub-sistemas. A análise dos resultados e discussão sobre futuros projetos que podem ser desenvolvidos utilizando este como base estão no fim deste documento.

2 Metodologia

Na sequência, é apresentada a descrição da metodologia empregada para o estudo de cada área deste projeto, assim como a viabilidade e interesse comercial.

2.1 Pesquisa

Para embasar este projeto, foi feita uma pesquisa inicial, abrangendo os seguintes assuntos:

- Sensores para captura de impressão digital: foram analisados diversos tipos de sensores (ótico, capacitivo, ultrassônico, etc) e suas vantagens e desvantagens;
- Técnicas de tratamento de imagem: janelamento, normalização, binarização, Filtro de Gabor, afinamento, transformada discreta de Fourier etc;
- Técnicas de extração de minúcias da digital: foram estudadas diferentes técnicas (baseadas em minúcias, texturas, etc), incluindo suas vantagens e desvantagens;
- Formas de armazenamento de minúcias e comparação entre *templates* – mapa da impressão digital composto pelas minúcias extraídas;
- Componentes e bibliotecas já existentes, que poderiam ser utilizados no desenvolvimento dos algoritmos;
- Interfaces gráficas utilizadas em projetos de pesquisa que visam comparar diferentes sub-sistemas;
- Métricas de desempenho utilizadas para avaliação e comparação entre algoritmos de autenticação biométrica;
- Teoria e aplicação de técnicas adaptativas.

2.2 Levantamento de Demandas de Mercado

Para melhor compreensão das necessidades da área, foi contactada a empresa Control iD (<http://www.controlid.com.br/>), especializada em soluções de controle de acesso, não só por cartões e folhas de ponto, mas também por biometria como impressão digital.

De acordo com o Diretor de Tecnologia desta empresa, Albert Nissimoff, há uma grande carência no mercado brasileiro de bons algoritmos de autenticação. Atualmente, a Control iD compra um algoritmo europeu que realiza a extração de minúcias e o *matching*. A solução adquirida é avaliada como uma das melhores disponíveis no mercado e, por isso, é bastante custosa.

Segundo Nissimoff, a Control iD tem interesse em um algoritmo de reconhecimento biométrico de alta qualidade produzido no Brasil, pois acredita que teria um custo inferior às soluções europeias. O Diretor de Tecnologia da Control iD ressaltou que atualmente existem poucas organizações no Brasil que fornecem algoritmos de reconhecimento biométrico, e que os existentes não apresentam desempenho comparável às soluções dos competidores internacionais.

Dessa forma, verifica-se uma grande oportunidade de mercado para este projeto, reforçando a motivação inicial dos autores e estimulando trabalhos futuros relacionados ao tema.

2.3 Desenvolvimento

Para implementação deste projeto, a equipe se organizou de forma a segmentar o projeto nos principais tópicos de abordagem. Essencialmente, as principais áreas de estudo são: pré-processamento de imagem, extração de parâmetro relevantes, autenticação de usuário, métricas de desempenho e adaptatividade. Com base

nesses tópicos, estimou-se o tempo de realização das atividades a fim de se criar um cronograma viável.

Além disso, o desenvolvimento abordou também o estudo de tecnologias disponíveis, além do aprendizado da tecnologia escolhida. Visto que algumas ferramentas e linguagens utilizadas não eram familiares à equipe.

2.3.1 Organização do desenvolvimento do projeto

No desenvolvimento, o projeto foi dividido em módulos: pré-processamento da imagem de entrada, extração de parâmetros relevantes, *matching*, cálculo de métricas e aplicação de técnicas adaptativas.

Para desenvolvimento do projeto de acordo com as especificações descritas ao longo deste documento, foi necessário arquitetar o banco de dados e planejar o fluxo do sistema, para então planejar a interface gráfica.

A equipe organizou-se primeiro definindo o escopo do projeto e, depois, ordenando o desenvolvimento de cada módulo do sistema. Os primeiros meses de projeto foram destinados a estudo não somente dos principais tópicos abordados no projeto, como também das tecnologias disponíveis, a fim de escolher a mais recomendável para a aplicação proposta.

O desenvolvimento colaborativo foi suportado pela utilização do Git como gerenciador de versões de código.

Além disso, a equipe estipulou inicialmente reuniões semanais com o Professor Doutor João José Neto, orientador deste projeto. Esses encontros foram fundamentais para o correto direcionamento do estudo e do projeto. Na etapa de implementação, as reuniões foram reduzidas, já que as principais decisões de projeto haviam sido tomadas anteriormente.

2.3.2 Cronograma

A tabela a seguir ilustra o cronograma resumido seguido durante o projeto.

Etapa	Mar – Abr	Mai - Jun	Jul - Ago	Set - Out	Nov
Pesquisa Bibliográfica					
Especificação					
Implementação					
Testes					
Documentação					

3 Conceitos e tecnologias

3.1 Sensores de captura de impressões digitais

Existem diferentes tipos de sensores que podem ser utilizados para a captura da impressão digital. A seguir há uma descrição dos principais.

A integração dos algoritmos de autenticação com o hardware de captura da digital não faz parte do escopo deste projeto, porém os diferentes dispositivos foram estudados com o objetivo de embasar a escolha de um dos bancos de dados de imagens fornecidas pela *Fingerprint Verification Competition (FVC)*.

3.1.1 Ultrassônico

O sensor de captura de impressão digital ultrassônica atua emitindo um feixe ultrassônico em direção à superfície do dedo, a fim de medir diretamente a profundidade dos vales com base no sinal refletido.

As condições de oleosidade da pele não afetam a imagem obtida, que reflete adequadamente a topologia dos vales. Contudo, estas unidades tendem a ser grandes e a requerer um alto tempo de processamento.

3.1.2 Ótico

O leitor ótico consiste basicamente em um sensor que captura uma imagem da impressão digital. A técnica mais usada para a aquisição desta imagem é a "*Frustrated Total Internal Reflections*" (*FTIR*), que será explicada a seguir.

Nesta técnica, usualmente o leitor óptico é composto por um vidro em formato de prisma, um espelho, uma luz e um processador. Quando o dedo toca a parte de cima do vidro prismático, as rugas entram em contato com a superfície do prisma, mas os vales permanecem a certa distância. Um dos lados do prisma é iluminado por uma luz difusa. A luz que penetra o prisma é refletida até os vales e absorvida pelas rugas. A falta de reflexão permite às rugas serem distinguidas dos vales. Essa imagem produzida é refletida por um espelho e então enviada ao processador que registra a imagem da digital que pressionou o vidro.



Figura 2 - Sensor Óptico

Fonte: <http://www.ligimports.com.br/persona-u4000>

3.1.3 Sensores de Estado Sólido

Esta classe de sensores é implementada através de uma matriz de pequenos sensores. Eles podem ser de 4 tipos: capacitivo, térmico, campo elétrico e piezoelétrico. Cada tipo está descrito a seguir.

Capacitivo: o sensor é formado por uma matriz de pequenos capacitores. A partir da capacitância em cada ponto da matriz, o sensor é capaz de identificar as cristas e vales da impressão digital. Uma das placas do capacitor é interna ao sensor e o dedo faz o papel da segunda placa. Onde há crista, o dielétrico é composto apenas pela superfície do sensor, e onde há um vale, o dielétrico é formado pela superfície e pela pequena quantidade de ar abaixo do vale. Como principal desvantagem, esse

sensor é bastante sensível, podendo ser danificado com facilidade, além de apresentar custo superior aos demais sensores.

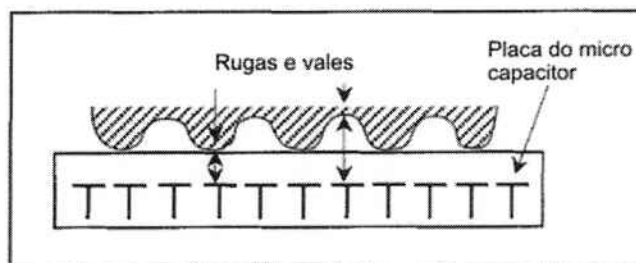


Figura 3 - Sensor Capacitivo

Fonte: http://www.gta.ufrj.br/grad/09_1/impdigital/index_files/Page298.htm

Térmico: consiste em uma matriz de pequenos termômetros. A superfície de contato é mantida a uma temperatura conhecida e após o toque do dedo a temperatura da superfície varia onde há cristas da digital, enquanto a região de vales sofre uma variação menor, permitindo que elas sejam distinguidas. Esse sensor exige termômetros de alta precisão, o que é pouco disponível no mercado.

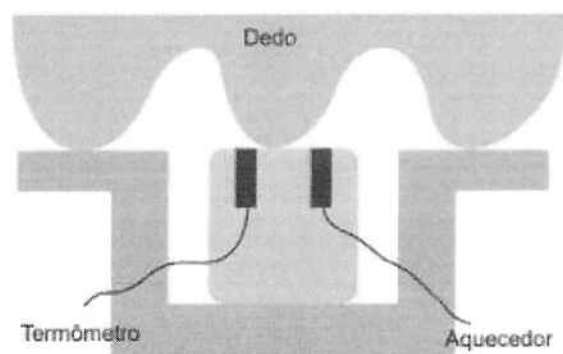


Figura 4 - Sensor térmico

Fonte: http://www.gta.ufrj.br/grad/09_1/impdigital/index_files/Page298.htm

Campo elétrico: consiste em uma matriz de pequenas antenas. Um sinal eletromagnético é enviado pela pele do usuário e é modulado por variações na pele, os vales e cristas. Esse sinal é detectado pelas micro antenas, sendo que as cristas geram uma modulação diferente da dos vales. A imagem da detecção por campo

elétrico é de altíssima qualidade, porém o método é bastante sensível a interferências eletromagnéticas.

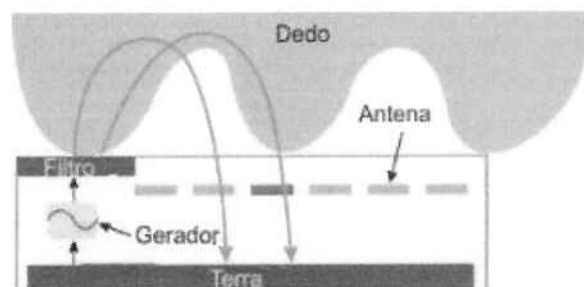


Figura 5 - Sensor de campo elétrico

Fonte: http://www.gta.ufrj.br/grad/09_1/impdigital/index_files/Page298.htm

Piezoelétrico: os pixels são realizados na forma de micro sensores de pequenas variações de pressão. Essas variações geram campos elétricos. As rugas aplicam uma pressão sobre o material, enquanto que os vales não, dessa forma é possível distingui-los. A principal vantagem deste sensor é dispensar a necessidade de dedo ou sensor limpos, porém apresenta como desvantagens a susceptibilidade a ruídos e a instabilidade de suas propriedades em função de variações de temperatura.

3.2 BDs de imagens de digitais

Os estudos das imagens provenientes de cada tipo de sensor de captura de impressão digital permitiram concluir que, dentre os diferentes tipos, os óticos são os que apresentam melhor custo-benefício, pois além de serem os mais utilizados e acessíveis atualmente, também apresentam uma imagem de boa qualidade.

Dessa forma, optou-se pelo banco de dados DB1 disponibilizado pela *Fingerprint Verification Competition* realizada em 2004 (<http://bias.csr.unibo.it/fvc2004/>). Este repositório de imagens é composto por imagens de 500 dpi, geradas por sensores óticos.

3.3 Classificação de Digitais

As impressões digitais são classificadas de acordo com o padrão de desenho das cristas, podendo ser classificadas como:

- Volta (*Loop*);
- Arco (*Arch*);
- Espiral (*Whorl*);
- Ou combinações de mais de um dos formatos citados.

A Figura 6 ilustra alguns dos formatos de digitais. Cada um desses formatos pode ter variações, como volta para esquerda, volta para a direita, entre outros.

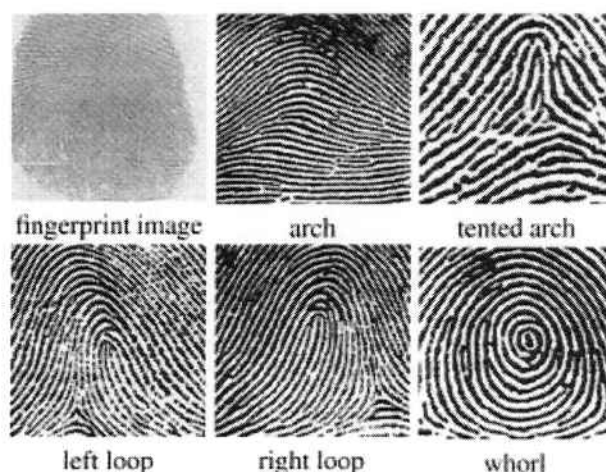


Figura 6 - Classificação de Tipos de Digitais

Fonte: (JAIN, HONG, 1998)

É possível indexar um banco de dados de impressões digitais pelo formato da digital, reduzindo o escopo de busca e, portanto, agilizando o processamento.

Essa é uma técnica interessante, apresentada aqui com fins didáticos, pois não houve necessidade de aplicá-la neste projeto, que não se propõe a trabalhar com volumes muito elevados de dados.

3.4 Reconhecimento de impressões digitais por minúcias

Existem diferentes formas de fazer reconhecimento por impressão digital, sendo a mais comum delas o reconhecimento baseado em minúcias, que foi a técnica adotada neste projeto.

Para extrair minúcias é necessária uma imagem de alta qualidade. Como as imagens originalmente capturadas nem sempre apresentam qualidade suficiente, muitas vezes esse método requer diversas etapas para tratamento da imagem, com o fim de melhorar sua qualidade, deixando-a mais nítida e eliminando ruídos. Geralmente, faz-se uso do(s) “core(s)” e do(s) “delta(s)” do dedo para definir a posição do dedo na imagem e o formato da digital (volta, arco, espiral ou combinação – exemplificados na Figura 6). Dessa forma, a identificação entre digitais é independente da angulação do dedo na imagem.

A Figura 7 demonstra o que são *core* e *delta* de uma impressão digital.

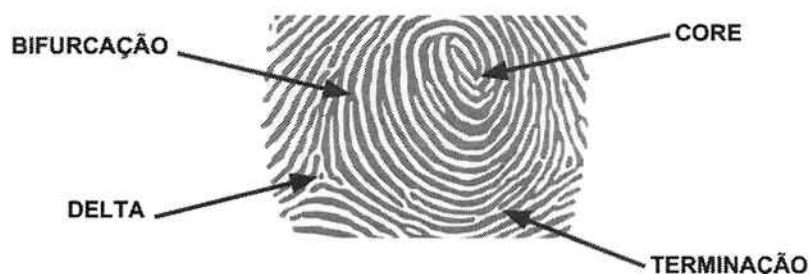


Figura 7 - Minúcias

Fonte: Autor

A etapa seguinte é a extração de minúcias. As minúcias mais comumente armazenadas como atributos de impressão digital são as terminações e bifurcações, como se pode observar na Figura 7. Há diversas formas para armazenamento das minúcias. A seguir, faz-se uma breve descrição de algumas delas.

Dado que o posicionamento das minúcias é único para cada indivíduo, gera-se um mapa que descreve a posição dessas minúcias e então esse mapa é comparado

com o de novas digitais de entrada para verificar se se trata de uma pessoa cadastrada no sistema ou não. O mapa gerado com as minúcias é comumente chamado *template*.

Como se observa na Figura 8 - Mapa de Minúcias, a última figura da direita corresponde ao mapa criado com as posições das minúcias encontradas na imagem. A cada vez que uma pessoa colocar a digital para autenticação no sistema, gera-se um novo mapa a partir da imagem de entrada obtida e o compara com os demais mapas já armazenados no sistema.

A comparação entre os mapas é realizada de acordo com a posição de cada minúcia na digital e determina quantitativamente, através de uma métrica adequada, quanto os *templates* são similares. Quanto mais parecidos, maior será a probabilidade de se tratar da imagem de um mesmo dedo.



Figura 8 - Mapa de Minúcias

Fonte: <http://www.intechopen.com/books/biometrics/minutiae-based-fingerprint-extraction-and-recognition>

Outra forma de autenticação de usuários é a partir da angulação entre as minúcias da impressão digital. Gera-se uma árvore com o mapa de angulação entre minúcias. Essa árvore é gerada quando se cadastra uma pessoa no sistema e é novamente gerada quando um indivíduo tenta autenticar-se no sistema. Os mapas gerados são comparados e, de acordo com a semelhança entre o novo mapa e os cadastrados no sistema, a pessoa pode ser autenticada ou não. A figura a seguir exemplifica a árvore gerada para comparação entre impressões digitais.



Figura 9 - Árvore de Minúcias

Fonte: http://www.novachron-zeiterfassung.de/en/fingerprint_recognition.php

A imagem da impressão digital de um mesmo indivíduo pode apresentar variações devido a ferimentos, cicatrizes, resíduos ou até mesmo à má qualidade da imagem. Essas variações resultam na identificação de falsas minúcias e, para isso, há diversos métodos que auxiliam na eliminação de falsas minúcias.

A Figura 10 exemplifica a redução da quantidade de minúcias extraídas após se aplicar algoritmo que elimina falsas minúcias, ou seja, minúcias que não correspondem a uma real bifurcação ou terminação de uma crista da impressão digital.

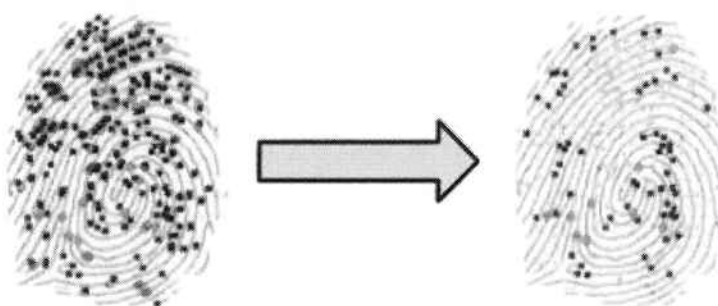


Figura 10 - Eliminação de Falsas Minúcias

Fonte: Autor

3.5 Reconhecimento de Padrões

Reconhecimento de Padrões (RP) busca classificar objetos em categorias ou classes (THEODORIDIS et al., 2006). Dados de entrada são selecionados a fim de extrair os atributos relevantes ao reconhecimento de padrão. A partir de regras, é possível utilizar esses atributos para a classificação dos objetos.

O processo para reconhecimento de padrão é composto pelas seguintes etapas:

- Aquisição dos dados: Obtenção e armazenamento dos dados de entrada;
- Pré-processamento: Tratamento dos dados adquiridos, removendo ruídos e inconsistências;
- Extração de características: Seleção de atributos relevantes presentes nos dados pré-processados, que permitam a identificação de um subconjunto;
- Classificação: Características extraídas são analisadas e de acordo com os modelos existentes, o objeto de entrada é classificado. Por exemplo, os padrões nas asas de abelhas variam conforme sua espécie. Dessa forma, a partir da extração e combinação de características da asa da abelha, é possível identificar a sua espécie (BUENO, 2010).

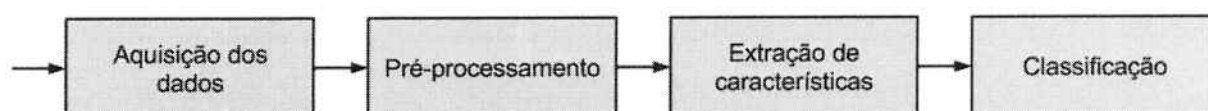


Figura 11- Diagrama do processo de reconhecimento de padrão

Fonte: Autor

Essencialmente, essas são as principais fases do reconhecimento de padrão. Porém, é comum o reconhecimento de padrão estar atrelado ao aprendizado de máquina. Nesta situação, há duas formas: (1) Inicialmente a máquina passa pela fase de aprendizado/treinamento, em que os atributos relevantes são selecionados e classificados em padrões e, posteriormente, o sistema inicia o reconhecimento de padrões de acordo com o que foi aprendido; (2) Embora não haja fase de treinamento, ao longo do processamento, cada etapa do sistema recebe um *feedback* da etapa seguinte, de modo a modificar o seu comportamento referente às classificações dos objetos de entrada durante o processamento.

No projeto desenvolvido, utiliza-se a forma (2) de reconhecimento de padrão. Com o *feedback* fornecido pelo operador do sistema, a aplicação muda seu comportamento de forma transparente ao usuário. Detalha-se esta etapa do projeto na seção 4.12.3.

3.6 Adaptatividade

A adaptatividade é uma característica associada ao comportamento automodificável de um sistema computacional. Este comportamento, autônomo, ocorre em resposta a estímulos de entrada e ao histórico de operação desses sistemas (NETO, 2001).

A tecnologia adaptativa corresponde ao conjunto de ferramentas, métodos e técnicas, que permitem solucionar problemas práticos utilizando modelos baseados em dispositivos adaptativos. Em seu modelo geral, um dispositivo adaptativo é definido como um conjunto finito de regras que pode sofrer modificações dinamicamente (NETO, 2001).

Existem diversas áreas de aplicações de adaptatividade e de tecnologia adaptativa:

- Inteligência computacional: aprendizagem de máquina, representação e manipulação do conhecimento;
- Computação natural, evolutiva e bio-inspirada;
- Sistemas de computação autonômica e reconfigurável;
- Processamento de linguagem natural, sinais e imagens: aquisição, análise, síntese, reconhecimento, conversões e tradução;
- Inferência, reconhecimento e classificação de padrões;
- Modelagem, simulação e otimização de sistemas inteligentes de: tempo real, segurança, controle de processos, tomada de decisão, diagnóstico, robótica;
- Simulação, arte por computador e jogos eletrônicos inteligentes;
- Outras aplicações da Adaptatividade, nas diversas áreas do conhecimento: ciências exatas, biológicas e humanas.

A Figura 12 ilustra a relação entre alguns dos principais assuntos na área de tecnologia adaptativa.

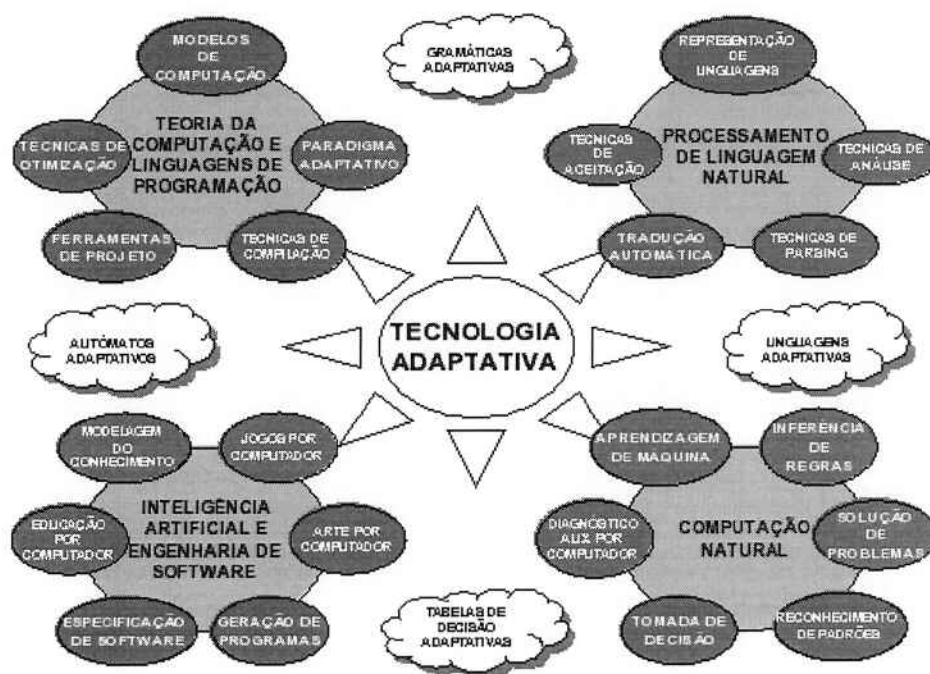


Figura 12 - Tecnologia Adaptativa

Fonte: Laboratório de Tecnologias Adaptativas – EPUSP (www.lta.poli.usp.br)

Este projeto aplica técnicas de adaptatividade ao reconhecimento de padrões.

4 Especificação do Sistema

Neste projeto, foram implementados três sub-sistemas de reconhecimento por impressão digital, com o objetivo de comparar esses sistemas no âmbito da qualidade das decisões e do tempo de processamento. Cada sub-sistema desenvolvido é detalhado nas seções a seguir.

4.1 Métricas de desempenho

A comparação entre o desempenho dos três sub-sistemas desenvolvidos será feita tendo como base as quatro principais métricas utilizadas pela competição internacional *FVC*. A metodologia de testes da *FVC* e a empregada neste projeto estão descritas na seções 6.1 e 6.2, respectivamente.

Taxa de falsa aceitação (*FAR – False Acceptance Rate*): métrica que indica a frequência com que o sistema aceita pessoas não cadastradas. Para aplicações sem um propósito específico, este indicador é um valor idealmente zero ou significativamente baixo. Esta métrica é calculada da seguinte forma:

$$\frac{\text{falsas aceitações}}{\text{falsas rejeições} + \text{verdadeiras aceitações}}$$

Taxa de falsa rejeição (*FRR – False Rejection Rate*): indica a frequência com que usuários cadastrados são rejeitados, isto é, o sistema não reconhece a pessoa como usuária, apesar de ela estar cadastrada na aplicação. Esta métrica é calculada como:

$$\frac{\text{falsas rejeições}}{\text{falsas rejeições} + \text{verdadeiras rejeições}}$$

Taxa de erro equivalente (*EER – Equal Error Rate*): representa o equilíbrio entre a taxa de falsa aceitação e a taxa de falsa rejeição, ou seja, o valor em que ambas as

métricas se igualam. Graficamente, a ERR é o ponto de intersecção entre as curvas de FAR e FRR, como se observa na Figura 13.

O *EER* indica que a proporção de falsas aceitações é equivalente a de falsas rejeições.

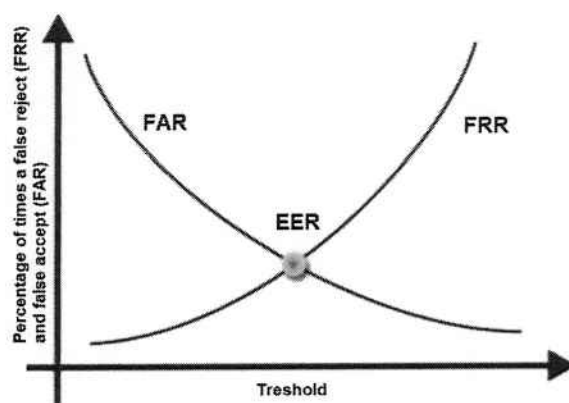


Figura 13 – Relação entre as principais métricas de desempenho (FAR, EER e FRR)

Fonte: <http://www.biometria.sk/en/principles-of-biometrics.html>

Muitos sistemas biométricos comercializados para aplicações civis buscam uma nota de corte que aproxime o comportamento do sistema ao *EER*, conforme pode ser visto na Figura 14 - Operação típica para diferentes aplicações, já que estes sistemas não costumam personalizar o desempenho conforme o contexto da aplicação, que é a proposta deste projeto.

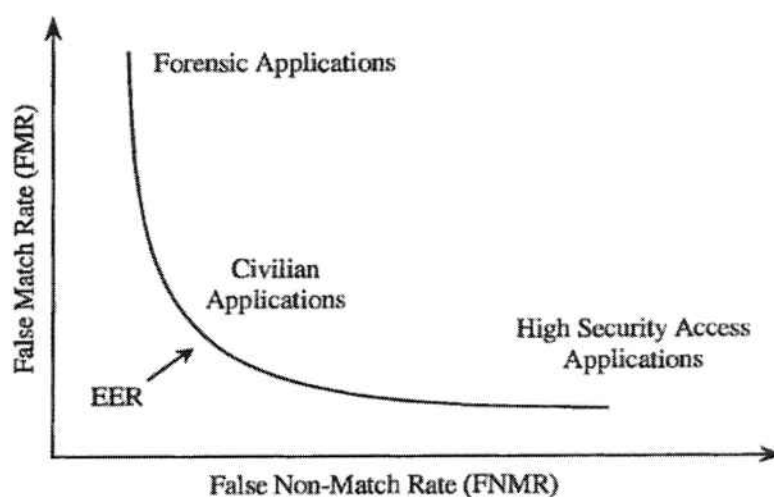


Figura 14 - Operação típica para diferentes aplicações

Fonte: MADDALA, S.; TANGELLAPALLY, S. R.: "Implementation and Evaluation of NIST Biometric Image Software for Fingerprint Recognition"

Tempo médio de processamento: Indica o tempo médio que o sistema leva para tomar uma decisão, seja de aceitação ou rejeição de um indivíduo. As métricas de resultado devem ser avaliadas em conjunto com esta métrica de tempo para permitir uma conclusão de qual sub-sistema possui melhor custo-benefício, conforme a necessidade de cada aplicação. Por exemplo, para um sistema que necessite alto nível de segurança, é preferível que a aplicação leve mais tempo para tomar a decisão do que a tome rapidamente, porém de forma errônea.

4.2 Sub-sistema 1: Sem pré-processamento e não adaptativo

Este sub-sistema é composto pelas etapas de obtenção da imagem, extração de minúcias e identificação de usuário. A Figura 15 demonstra o fluxo do algoritmo com suas principais etapas.



Figura 15 - Fluxo de execução do Sub-sistema 1

Fonte: Autor

A partir da imagem de entrada, o sub-sistema extrai as minúcias e gera o *template* referente àquela digital. A etapa seguinte é a comparação do *template* gerado com os armazenados no banco de dados, em que ocorre a avaliação se a digital de entrada é de um usuário cadastrado ou não.

É importante destacar que o Sub-sistema 1 não possui qualquer etapa de pré-processamento de imagem.

4.3 Sub-sistema 2: Com pré-processamento e não adaptativo

O Sub-sistema 2 utiliza técnicas convencionais para processamento de imagem e identificação de usuários.

Esta aplicação possui todas as funcionalidades do Sub-sistema 1, além de ter pré-processamento de imagem, conforme pode ser visto na Figura 16 - Fluxo de execução do Sub-sistema 2. A eficácia do Sub-sistema 2 nem sempre é superior ao 1, pois a aplicação da etapa de normalização de forma não adaptativa pode piorar a qualidade da imagem, de modo a dificultar a extração de minúcias e piorar a qualidade delas.

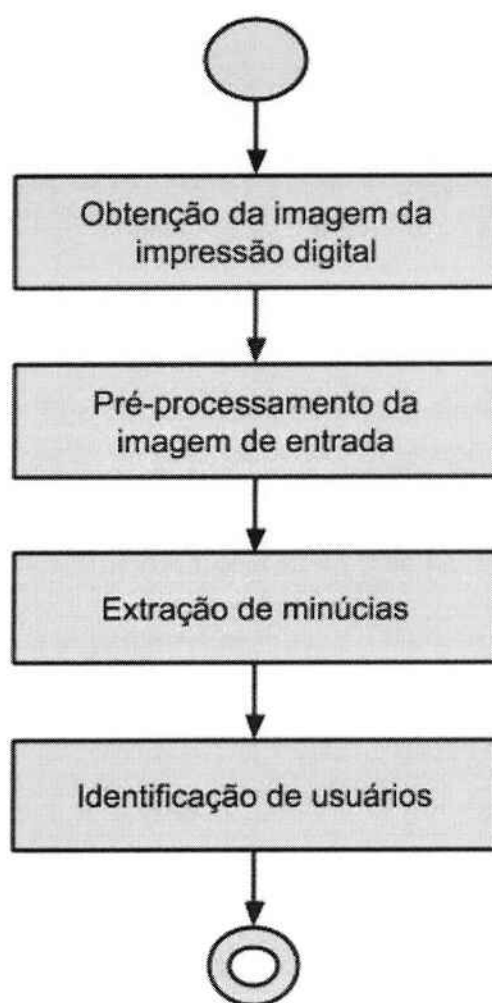


Figura 16 - Fluxo de execução do Sub-sistema 2

Fonte: Autor

4.4 Sub-sistema 3: Com pré-processamento e adaptativo

O Sub-sistema 3 é similar ao 2, porém com a aplicação de técnicas adaptativas, desde a etapa de pré-processamento da imagem de entrada até o *matching* – identificação de usuários. O fluxograma do algoritmo deste sub-sistema encontra-se na Figura 17 - Fluxo de execução do Sub-sistema 3. Vale ressaltar que esta aplicação é o foco da proposta deste projeto, já que incorpora métodos adaptativos.

Com a adaptatividade empregada no pré-processamento da imagem, a extração de minúcias é refinada, isto é, a identificação dos parâmetros da digital torna-se mais precisa, desprezando ruídos da imagem de entrada.

O Sub-sistema 3 apresenta também a possibilidade de modificar o seu comportamento de acordo com o nível de exigência da aplicação final. Os níveis de exigência disponíveis são: padrão, tolerante e rigoroso. O caso padrão ajusta a aplicação para uma nota de corte de similaridade entre *templates* pré-estabelecida; o caso tolerante adapta a nota de corte inicial de forma a reduzir a taxa de falsa rejeição; e o caso rigoroso adapta a nota de corte de forma a reduzir a taxa de falsa aceitação.

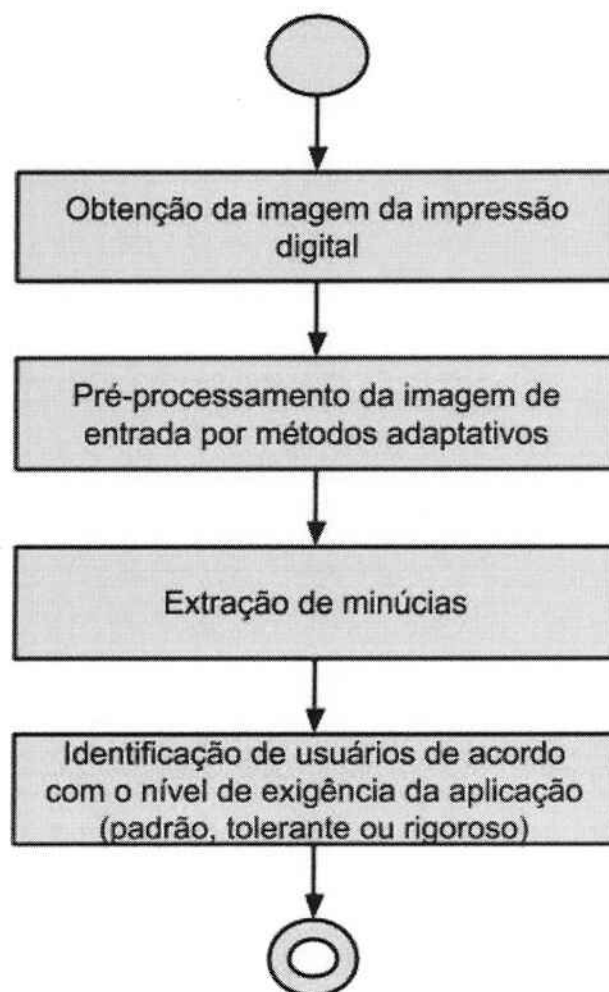


Figura 17 - Fluxo de execução do Sub-sistema 3

Fonte: Autor

Cada etapa que compõe os três sub-sistemas desenvolvidos é detalhada nas seções 4.9 a 4.12.

4.5 Escopo

Os três algoritmos de identificação biométrica por impressão digital são do tipo 1-N, isto é, a partir de uma digital de entrada, é feita a comparação com N digitais armazenadas no BD com a finalidade de verificar se aquela digital corresponde a algum dos usuários cadastrados.

Os algoritmos recebem como entrada uma imagem de 500 dpi e tomam uma decisão: de aceitação ou rejeição do usuário.

Como inovação, um dos algoritmos deve utilizar adaptatividade para permitir uma análise da potencial melhoria de desempenho em comparação com os algoritmos tradicionais.

A comparação é feita entre o algoritmo adaptativo e duas versões de algoritmo tradicional: uma simples, sem pré-processamento (Sub-sistema 1), e uma completa, porém sem a utilização de métodos adaptativos (Sub-sistema 2).

4.6 Requisitos Funcionais

Os requisitos funcionais são os mesmos para os três sub-sistemas:

1. Seleção de uma imagem de impressão digital armazenada no sistema para realização do *matching*;
2. Visualização de cada etapa de pré-processamento dos sub-sistemas 1, 2 e 3, sendo possível exibir as imagens alteradas e as métricas de:
 - a. Resultado do *matching* para cada sub-sistema;
 - b. Tempo de processamento total de cada sub-sistema;
 - c. Tempo de processamento de cada etapa;
 - d. Quantidade de minúcias extraídas;
 - e. Qualidade média das minúcias extraídas (valor obtido pelo componente MINDTCT do NBIS, descrito em 4.11.1).
3. Visualização das métricas gerais de desempenho dos sub-sistemas: *FAR* e *FRR*, conforme definidas pela competição de algoritmos de identificação biométrica, *Fingerprint Verification Competition (FVC)*.

4.7 Requisitos Não-Funcionais

Na sequência, listam-se os requisitos não-funcionais estabelecidos para o projeto:

1. Cada sub-sistema suporta até 1.000 digitais cadastradas;
2. O sistema processa imagens de digitais provenientes de leitores ópticos (são utilizadas imagens disponibilizadas pela FVC);
3. As métricas de desempenho devem ser calculadas de forma independente para cada sub-sistema e modo de operação, permitindo a comparação entre eles;
4. Para cada sub-sistema (1, 2 e 3), são armazenados *templates* de uma mesma imagem de digital, porém esses *templates* com as minúcias extraídas são diferentes para cada sub-sistema, visto que o processo de extração de minúcias é distinto entre eles;
5. O *matching* de cada sub-sistema é realizado de forma individual. Compara-se somente os *templates* extraídos pelo sub-sistema em questão. Esse requisito é necessário pois garante que cada sub-sistema atue de forma isolada, impedindo que suas métricas sejam influenciadas por outros sub-sistemas.

4.8 Estrutura do Banco de Dados

A estrutura do Banco de Dados (BD) foi refinada ao longo do projeto. O BD modelado inicialmente possuía a configuração exibida na Figura 18.

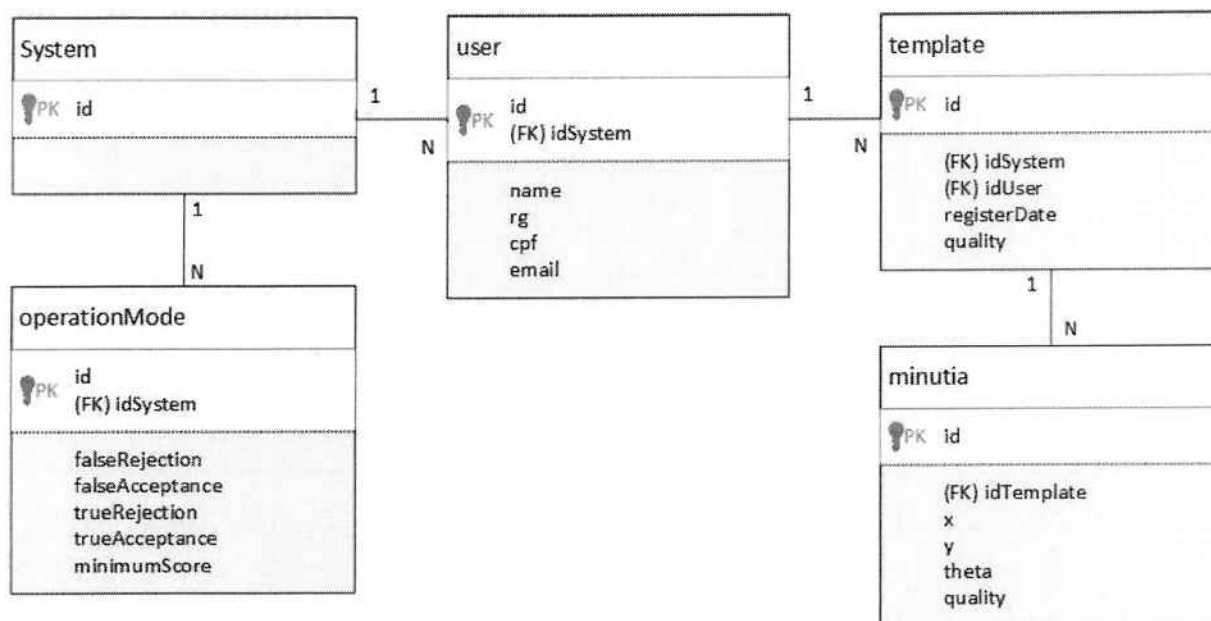


Figura 18 - Banco de Dados (versão inicial)

Fonte: Autor

Cada sub-sistema possui de um a muitos usuários, que possuem de um a muitos *templates*, que possuem de uma a muitas minúcias. Além disso, cada sub-sistema possui de um a infinitos modos de operação, que armazenam as métricas de desempenho daquele modo e a nota de corte mínima utilizada na decisão de aceitação ou rejeição de usuários.

Os usuários possuem dados de identificação pessoal como RG e CPF. Os *templates* possuem a data em que foram cadastrados e a qualidade média das minúcias que os compõem. Já as minúcias possuem coordenadas *x* e *y*, orientação *theta* e nota de qualidade.

Com essa estrutura de BD, para realizar o *matching* era necessário buscar todos os *templates* cadastrados e, para cada ID de *template*, buscar as minúcias relacionadas a ele e recriar um arquivo *.xyt*, que é o formato de entrada do BOZORTH3 (componente que gera a nota de similaridade, descrito na seção 4.12.1).

Na prática, essa arquitetura mostrou um baixa desempenho: o tempo de processamento ficou significativamente alto (cerca de 10 segundos por tentativa de acesso) e desagradável ao usuário. Isso aconteceu, pois cada *template* possui

muitas minúcias (ordem de algumas centenas) e cada minúcia devia ser tratada adequadamente para converter seus parâmetros em uma *String* no formato esperado pelo Bozorth3.

Por esse motivo, o Banco de Dados foi remodelado, de forma que todas as minúcias já estejam armazenadas em formato *String* em um dos campos da tabela *Template*, tornando imediata a geração do arquivo .xyt correspondente a cada template. Com essa modificação, a tabela *Minutia* tornou-se desnecessária.

A versão revisada do BD é a apresentada na imagem a seguir.

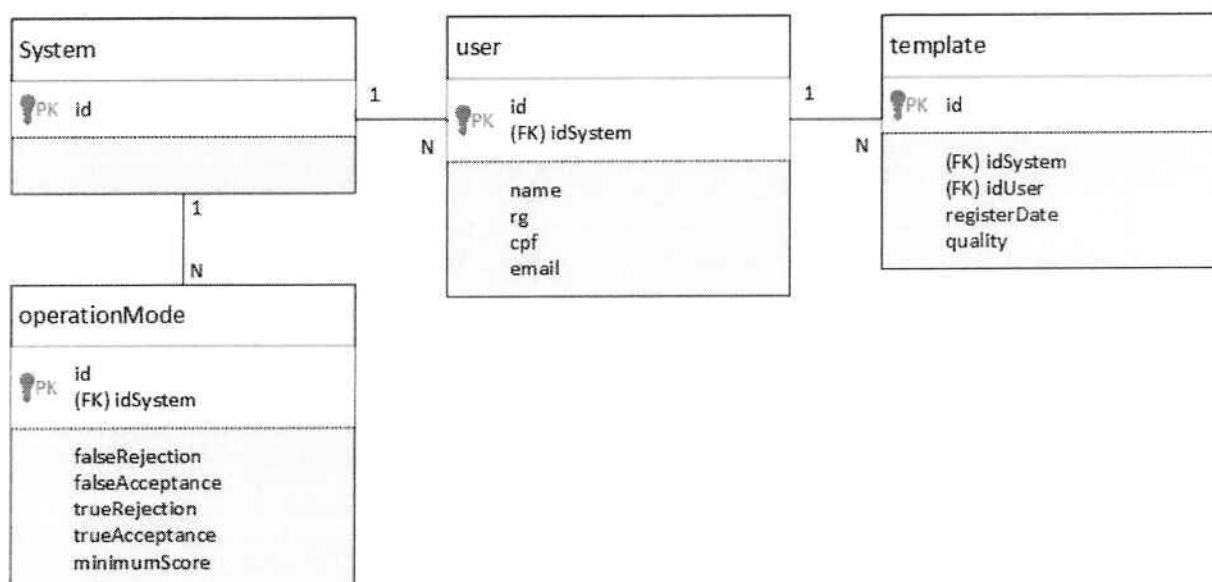


Figura 19 - Banco de Dados (versão final)

Fonte: Autor

Essa alteração se mostrou eficaz reduzindo o tempo de execução do algoritmo a 5% do tempo gasto anteriormente.

4.9 Imagem de entrada

Como imagem de entrada, seleciona-se uma das imagens disponíveis para escolha no sistema.

Como descrito na seção 3.2 - BDs de imagens de digitais, para cadastro no sistema, escolheu-se o banco de imagens DB1 da competição de 2004, por estar numa resolução adequada (500 dpi), além de serem imagens provenientes de sensor ótico, o mais utilizado dentre os sensores de captura de impressão digital.

Para o sistema desenvolvido, não há restrição referente à resolução da imagem, pois a etapa de janelamento, a ser descrita em 4.10.2, calcula as janelas com base na quantidade de dpi da imagem. Mas é recomendável que a imagem tenha pelo menos 500 dpi de resolução, pois isso garante uma qualidade melhor da figura obtida a ser tratada.

4.10 Pré-processamento

Conforme mencionado anteriormente, há três sub-sistemas que são executados simultaneamente com fins comparativos. Somente os sub-sistemas 2 e 3 possuem pré-processamento da imagem de entrada, o 3 é composto por mais etapas, visto que faz uso de técnicas adaptativas. Além disso, a etapa de normalização, comum aos dois sub-sistemas, é processada de forma distinta em cada um, pois no Sub-sistema 3 se aplica uma normalização adaptativa, como é descrito a seguir.

A Figura 20 representa as fases que compõem o pré-processamento de imagem para os sub-sistemas 2 e 3. As etapas coloridas em verde são etapas adaptativas. Na sequência cada uma das etapas é detalhada.

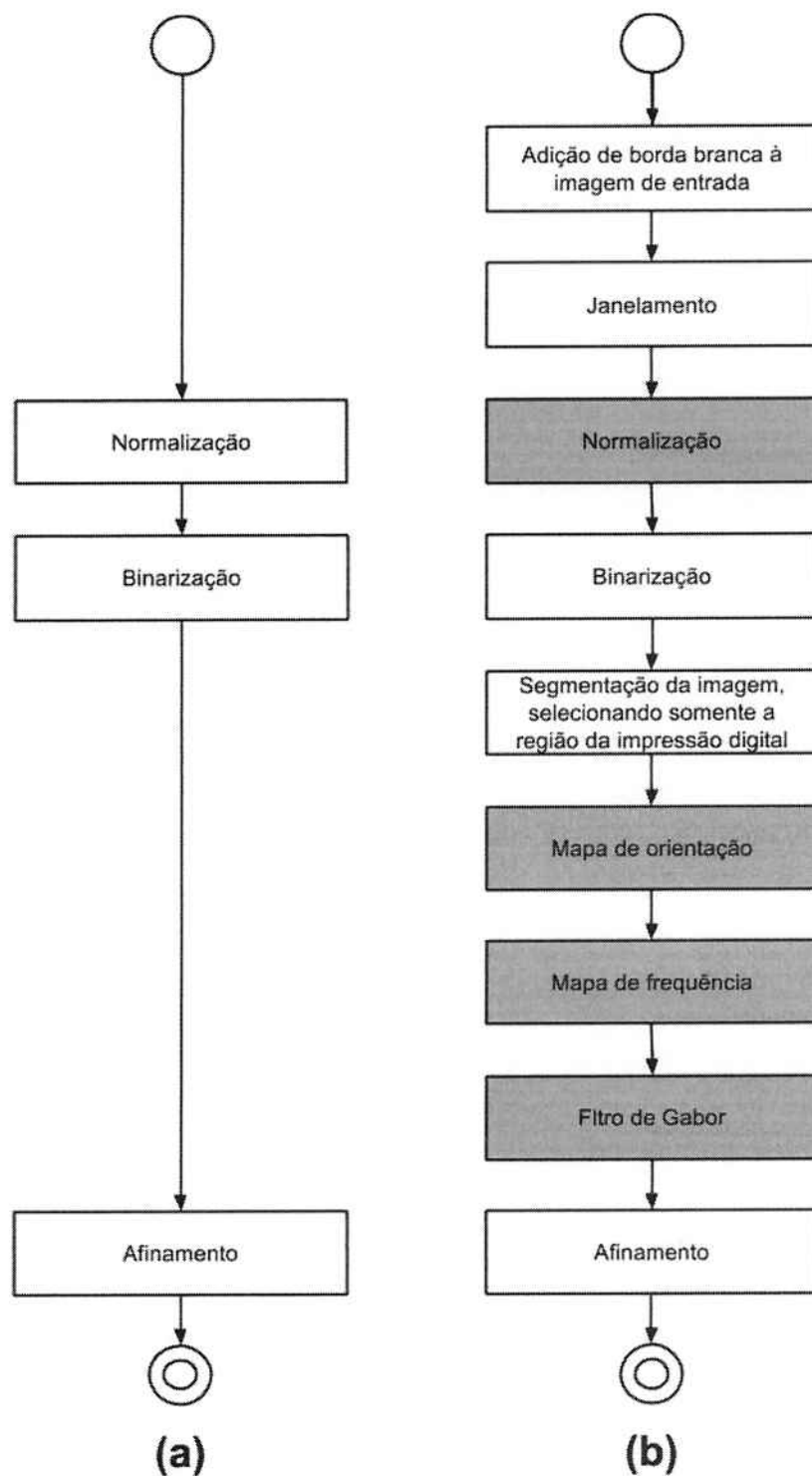


Figura 20 - Diagrama das etapas de pré-processamento da imagem de entrada dos sub-sistemas 2 e 3. (a) Sub-sistema 2 - com pré-processamento de imagem, mas sem adaptatividade -, (b) Sub-sistema 3 - com pré-processamento de imagem e com adaptatividade.

Fonte: Autor

4.10.1 Adição de borda

Sub-sistemas em que se aplica: 3.

Nesta etapa, acrescenta-se borda branca à imagem de entrada. Este estágio do sistema é necessário para que o janelamento, etapa seguinte, não despreze informações relevantes da impressão digital.

Adiciona-se borda branca nas laterais da imagem. A espessura da borda é calculada com base nas dimensões da imagem e em sua resolução (em dpi). Para o janelamento, é necessário que a imagem tratada tenha suas dimensões múltiplas de N , sendo a janela de dimensão $N \times N$. O cálculo de N é detalhado na seção 4.10.2.

É possível observar o resultado da adição de borda na Figura 21.

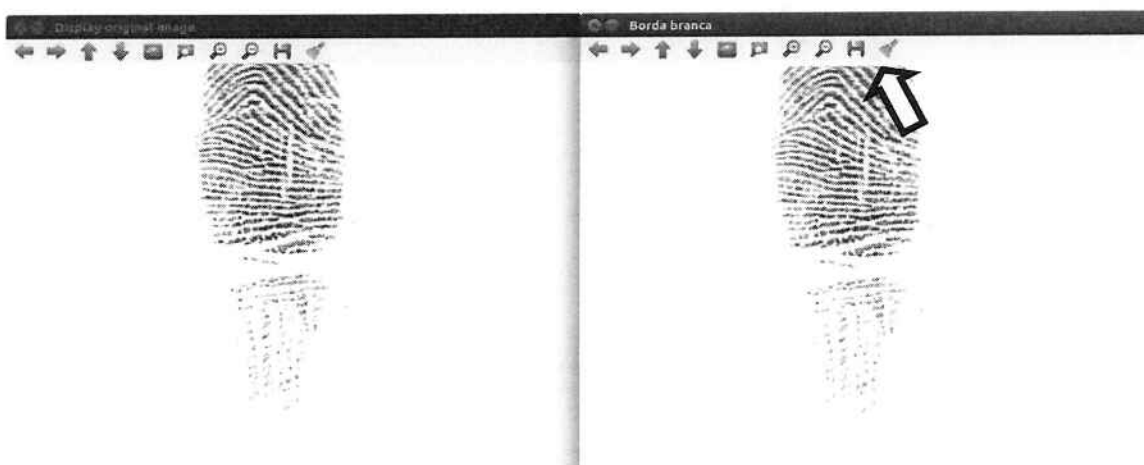


Figura 21 – Imagem original (sem bordas brancas), à esquerda; e com a adição de bordas brancas, à direita.

Fonte: Autor

4.10.2 Janelamento

Sub-sistema em que se aplica: 3.

Na etapa de janelamento, o tamanho da janela é calculado com base na definição da imagem (em dpi), então toda a imagem é segmentada em janelas, como se observa na Figura 22. Esta etapa é aplicada somente no sub-sistema adaptativo, sendo crucial visto que as etapas seguintes são executadas com base nas características individuais de cada uma das janelas. Isto é, os filtros, que são aplicados, adaptam-se às características de cada região da imagem, em vez de serem aplicados homogeneamente na imagem inteira.



Figura 22 – Janelas

Fonte: Autor

As janelas são salvas em uma estrutura de dados implementada em C++ que consiste em um vetor de vetores de ponteiros para "window", isto é, `vector < vector <window*> >`. Isso equivale a uma matriz em que cada posição corresponde a um ponteiro para a janela correspondente àquela posição. E dentro de cada janela ("window") temos os seguintes dados:

Window
quality: Int frequency: Double angle: Double imageWindow: Mat

Figura 23 - Estrutura de uma "window"

Fonte: Autor

O atributo *imageWindow* é do tipo *Mat* do OpenCV e armazena os pixels da janela. *Mat* é um tipo similar a uma matriz, em que cada posição armazena a informação de um pixel da figura.

Dada a importância do janelamento, vale ressaltar que o tamanho das janelas deve ser adequado, pois uma janela muito pequena não possui informações suficientes da digital, que permitam um bom resultado dos filtros; já uma janela muito grande faz

com que o filtro seja menos adaptativo e as particularidades de cada região não sejam tratadas adequadamente.

4.10.3 Normalização convencional (não-adaptativa)

Sub-sistema em que se aplica: 2.

Consiste em normalizar os tons de cinza de forma que a imagem inteira fique com melhor contraste. Nessa etapa procura-se corrigir a falta de nitidez da imagem.

A figura normalizada é definida por:

$$N(i, j) = M_0 + \sqrt{\frac{V_0 (I(i, j) - M)^2}{V}}, \text{ se } I(i, j) > M,$$

ou

$$N(i, j) = M_0 - \sqrt{\frac{V_0 (I(i, j) - M)^2}{V}}, \text{ se } I(i, j) \leq M.$$

em que:

- $I(i, j)$ é o valor do pixel (i, j) em escala de cinza;
- $N(i, j)$ é o valor do pixel (i, j) normalizado em escala de cinza;
- M a média da intensidade de todos $I(i, j)$ da imagem ou região selecionada;
- V a variância de todos $I(i, j)$ da imagem ou região selecionada;
- M_0 é a média desejada;
- V_0 é a variância desejada.

No método convencional, aplica-se a normalização somente uma vez para a imagem completa. Isso algumas vezes compromete a nitidez da imagem, tornando-a demasiadamente clara ou escura.

4.10.4 Normalização adaptativa

Sub-sistema em que se aplica: 3.

Com base na claridade média de cada janela da imagem, normalizam-se os tons de cinza. No sub-sistema adaptativo, a normalização é aplicada de forma individualizada em cada uma das janelas. Desta forma, a qualidade da imagem gerada é significativamente superior, e a figura da impressão digital torna-se visivelmente mais nítida.

Na Figura 24, observa-se a diferença entre a normalização aplicada de forma convencional e de forma adaptativa. No exemplo, a aplicação do filtro sem adaptatividade, apesar de melhorar a qualidade da imagem em algumas regiões, em outras piora significativamente, algumas cristas da impressão digital são agrupadas, como nota-se na região superior esquerda da imagem, dificultando a identificação das minúcias.



Figura 24 - (a) Imagem original, (b) normalização aplicação de forma convencional, (c) normalização aplicada adaptativamente (por janela)

Fonte: Autor

4.10.5 Binarização

Sub-sistemas em que se aplica: 2 e 3.

Na binarização todos os pixels são convertidos para preto ou branco de acordo com a intensidade de cinza que possuem. Basicamente, os pixels com intensidade de cinza maior ou igual a 127 são convertidos para 0 (preto), já pixels com valor superior a 127 são convertidos para 255 (branco). Esta etapa é necessária, pois no momento de extração de minúcias, a imagem será comparada com máscaras que representam os padrões de minúcias em branco e preto. Na Figura 25 a seguir observa-se o resultado da binarização:

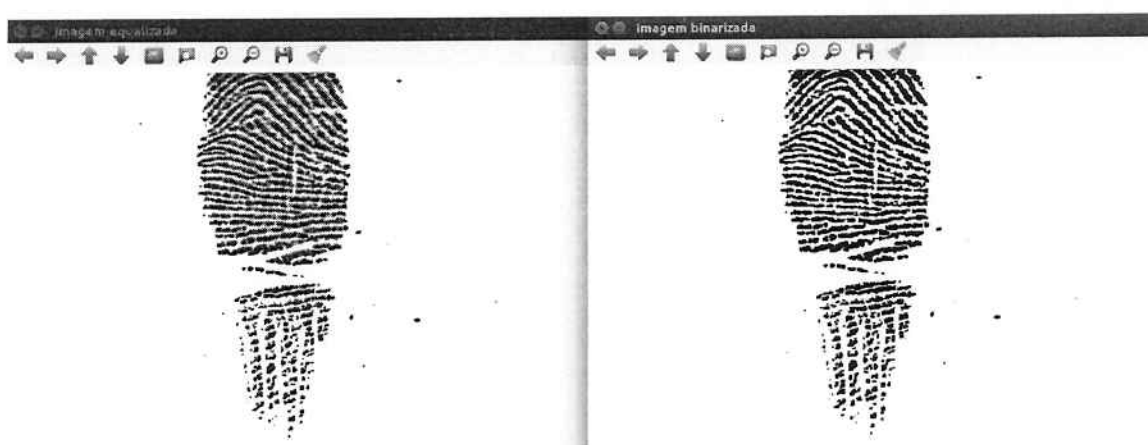


Figura 25 - Binarização

Fonte: Autor

4.10.6 Mapa de orientação

Sub-sistema em que se aplica: 3.

Nesta etapa, a direção das cristas de cada janela é avaliada e uma direção que aponta no sentido do gradiente é definida. O gradiente aponta perpendicularmente

às cristas, que podem ser vistas como análogas a curvas de nível. O mapa de orientação é necessário, pois a ortogonal à orientação de cada janela é um dos parâmetros do Filtro de Gabor.

Para obtenção do ângulo de orientação das cristas, utiliza-se o operador de Sobel. Apesar da biblioteca OpenCV possuir este operador implementado, sua utilização não apresentou resultados satisfatórios, por isso calculou-se o ângulo de forma clássica, como é descrito a seguir.

O operador de Sobel é um possível método para o cálculo do gradiente da imagem. Nesse operador, a aproximação da função gradiente é realizada através de uma matriz máscara onde os elementos representam valores espaciais da função Gradiente de Sobel a ser aplicada através do processo de convolução bidimensional (CASTRO, 2008).

Comumente, as máscaras para cálculo dos gradientes são 3x3. Segundo Nixon e Aguado, a virtude da utilização do operador de Sobel estendido, que possui máscaras 5x5, é a amenização dos ruídos da imagem. Por isso, adotou-se o operador estendido.

-1	0	1
-2	0	2
-1	0	1

(a)

1	2	1
0	0	0
-1	-2	-1

(b)

-2	-1	0	1	2
-2	-4	0	4	2
-2	-4	0	4	2
-2	-4	0	4	2
-2	-1	0	1	2

(c)

-2	-2	-2	-2	-2
-1	-4	-4	-4	-1
0	0	0	0	0
1	4	4	4	1
2	2	2	2	2

(d)

Figura 26 - Máscaras de Sobel 3x3 e 5x5, (a) e (c) para obtenção do gradiente horizontal, (b) e (d) para obtenção do gradiente vertical

Fonte: Autor

Primeiramente, os pixels da imagem são transformados em binário (1 ou 0), ou seja, os pixels brancos – de valor 255 – são alterados para 1 e os pretos mantidos em 0 (zero). Vale lembrar que a imagem já havia sido binarizada anteriormente, como descrito na seção 4.10.5, por isso não há pixels em tons de cinza.

Com a matriz de pixels da imagem preenchida somente com valores 1 e 0 (zero), máscaras do operador de Sobel são aplicadas para obter o gradiente horizontal ($\delta x(i, j)$) e o gradiente vertical ($\delta y(i, j)$) para cada janela (i, j) . Calculados os gradientes, estima-se a orientação local de cada janela com (KULKARNI et al., 2008):

$$V_x(i, j) = \sum_{u=1}^w \sum_{v=1}^w 2 \delta_x(u, v) \delta_y(u, v)$$

$$V_y(i, j) = \sum_{u=1}^w \sum_{v=1}^w (\delta_x^2(u, v) - \delta_y^2(u, v))$$

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

Obtida a orientação θ de cada janela, basta rotacionar 90° para obter-se a normal à orientação da janela, que é um dos parâmetros utilizados no Filtro de Gabor.

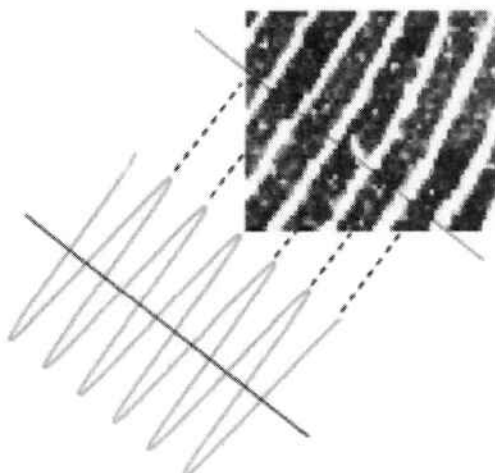
4.10.7 Mapa de frequência

Sub-sistema em que se aplica: 3.

Consiste no cálculo da frequência média de cristas da impressão digital para cada janela da imagem. O mapa de frequência é necessário, pois gera um dos

parâmetros necessários para a aplicação do Filtro de Gabor, que é detalhado na seção 4.10.8.

Após a seleção da imagem da janela, projetam-se os valores de cinza de todos os pixels localizados em cada janela ao longo da direção ortogonal à orientação calculada anteriormente (THAI, 2003). A projeção forma uma onda 1D com os extremos locais correspondendo às cristas e vales da impressão digital (RAICEVIC et al., 2009). A Figura 27 exemplifica a forma de onda obtida a partir da frequência de cristas de uma janela.



*Figura 27 – Modelagem de cristas e vales como um onda de formato senoidal
(ALONSO-FERNANDEZ et al., 2003)*

Para cada janela (i, j) , calcula-se $K(i, j)$, número médio de pixels entre dois picos consecutivos da onda 1D gerada. A frequência $f(i, j)$ é computada como $f(i, j) =$

$$\frac{1}{K(i, j)}.$$

No Sub-sistema 3, o cálculo da frequência de cristas é feito com auxílio da biblioteca OpenCV, que disponibiliza a função que aplica a transformada discreta de Fourier (*Discrete Fourier Transform – DFT*). O uso de transformadas permite observar características de um sinal que já estavam presentes nele, porém não podiam ser observadas nesse domínio. Neste projeto, leva-se o sinal do domínio do espaço para o domínio da frequência.

Na sequência, observa-se a aplicação deste filtro em uma imagem comparável a uma janela da impressão digital.

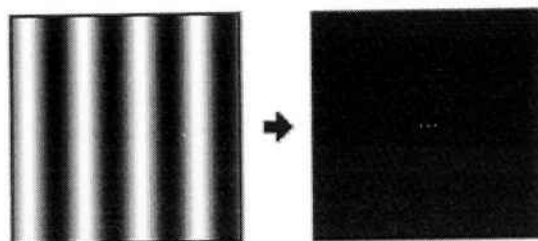


Figura 28 - DFT aplicada a imagem com 4 ciclos

Fonte: http://www.imagemagick.org/Usage/fourier/#fft_partial

A aplicação da *DFT* na imagem gera um vetor de números complexos, que correspondem às componentes senoidais da imagem. Pode-se extrair a magnitude desses valores complexos, colocar as magnitudes em escala logarítmica e “plotá-las” como pixels em uma nova imagem, como exemplificado na Figura 28. Neste caso, a imagem obtida é composta por três pontos: o ponto central, que representa a componente DC, e os demais, que representam a variação no sinal gerado a partir da transformação da imagem da janela. Estes dois pontos não centrais estão afastados 4 pixels em relação ao central. Isso significa que há 4 ciclos completos do sinal na imagem de entrada. Com essa informação, e sabendo o tamanho da imagem de entrada, é possível calcular tanto a frequência quanto o comprimento de onda.

A seguir há outro exemplo para comparação, em que a frequência é maior e, conseqüentemente, o espaçamento entre os pontos, representando o número de ciclos, também é maior.

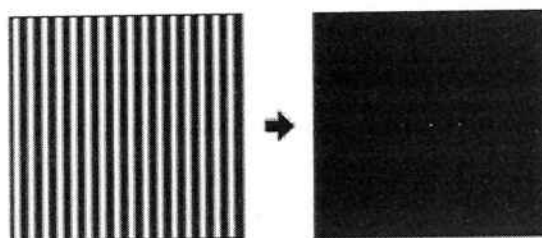


Figura 29 - DFT aplicada a imagem com 16 ciclos

Fonte: http://www.imagemagick.org/Usage/fourier/#fft_partial

O principal desafio ao utilizar a *DFT* é na aplicação de imagens pouco padronizadas ou com ruídos, como é o caso de algumas janelas da impressão digital, principalmente as que possuem o *core* ou *delta*. Nesses casos, a aplicação da *DFT* resulta em inúmeras componentes de frequência, geralmente em diversas direções, como se observa na Figura 30.

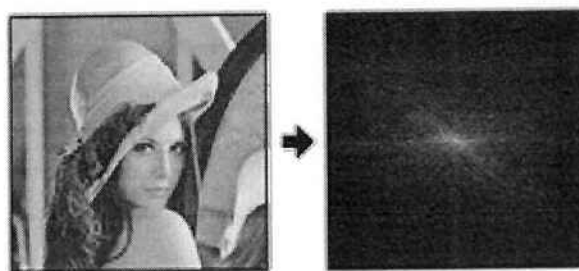


Figura 30 - DFT aplicada a imagem complexa

Fonte: http://www.imagemagick.org/Usage/fourier/#fft_partial

4.10.8 Filtro de Gabor

Sub-sistema em que se aplica: 3.

Filtro de Gabor é um filtro passa-banda, uma combinação de uma função harmônica com uma função gaussiana.

Neste projeto, foi utilizado um filtro de Gabor adaptativo: cada janela da imagem é filtrada de forma diferente com base na frequência e orientação daquela janela.

Matematicamente, o filtro de Gabor pode ser descrito da seguinte maneira:

$$g(x, y; \theta, f) = \exp \left\{ -\frac{1}{2} \left(\frac{x_{\theta}^2}{\sigma_x^2} + \frac{y_{\theta}^2}{\sigma_y^2} \right) \right\} \cdot \cos(2\pi \cdot f \cdot x_{\theta})$$

Em que:

- f é a frequência de cada janela;
- σ_x e σ_y são os valores das variâncias do sinal nas direções x e y , respectivamente;
- $[x_\theta, y_\theta]$ são os pontos $[x, y]$ rotacionados por θ . Essa rotação é calculada da seguinte forma:

$$\begin{pmatrix} x_\theta \\ y_\theta \end{pmatrix} = \begin{pmatrix} \sin \theta & \cos \theta \\ -\cos \theta & \sin \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

O Filtro de Gabor é muito importante na eliminação de ruídos da imagem que resultariam em falsas minúcias. Ele analisa o padrão da imagem e corrige regiões que apresentam desvios em relação a esse padrão. Por exemplo, para uma dada frequência de cristas, se existirem pixels de ruído fazendo com que a frequência seja maior numa pequena parte da imagem, o ruído é eliminado.

A seguir uma ilustração da imagem resultante da aplicação do Filtro de Gabor:



Figura 31 - Filtro de Gabor

Fonte: http://openi.nlm.nih.gov/detailedresult.php?img=3338710_pone.0035704.g002&req=4

4.10.9 Afinamento

Sub-sistemas em que se aplica: N/A*.

**Etapa apresentada para fins didáticos, pois a extração de minúcias, da forma como foi implementada neste projeto, dispensa a etapa de afinamento. Detalhes na seção 4.11.1.*

Nessa última etapa do pré-processamento, as cristas são afinadas até possuírem largura de 1 pixel. Isso pode ser necessário, pois a extração de minúcias compara a imagem com máscaras que representam padrões de minúcias e essas máscaras podem ter sido definidas apenas para o caso de cristas com 1 pixel de espessura. Abaixo está ilustrado o efeito da afinamento.



Figura 32 - Afinamento

Fonte: http://www.griaulebiometrics.com/images/books/understanding/File_028.png

A etapa de afinamento foi desenvolvida no projeto, porém não é utilizada visto que o componente MINDTCT, utilizado para a extração de minúcias, possui máscaras que toleram espessura de cristas superior a 1 pixel. Isso é detalhado na seção que segue, 4.11.1.

4.11 Extração de minúcias

Extração de minúcias é a etapa em que são identificados parâmetros relevantes e que tornam possível distinguir cada indivíduo, como descrito na sessão **Error! Reference source not found.**

É importante lembrar que a utilização de uma imagem livre de ruídos aumenta a probabilidade de a extração de minúcias ser condizente com a realidade, isto é, somente as reais minúcias da impressão digital são identificadas, desprezando os ruídos e impurezas da imagem.

4.11.1 Identificação de candidatas a minúcias

Sub-sistemas em que se aplica: 1, 2 e 3.

Após o tratamento da imagem de entrada, as minúcias são identificadas e extraídas. Neste projeto, é feita apenas a extração de terminações e bifurcações, que são as minúcias mais comuns e suficientes para o reconhecimento de um indivíduo.

Para a extração, utiliza-se o MINDTCT, que é um componente *open source* do *NIST Biometric Image Software (NBIS)*. As principais funcionalidades do *NBIS* são detalhadas em 5.7.

Essa decisão de projeto foi tomada após consultar o especialista Albert Nissimoff da empresa Control iD, que recomendou a utilização do componente MINDTCT devido ao seu bom desempenho na extração de minúcias. Ele destacou apenas a importância de enviar uma imagem de boa qualidade como entrada, pois o tratamento interno desse componente é limitado.

O algoritmo executado internamente pelo MINDTCT obtém a posição e a orientação das minúcias. Para localizar os parâmetros, o algoritmo compara a matriz de pixels da imagem binarizada com máscaras que representam os padrões de uma terminação ou de uma bifurcação (ilustração a seguir).

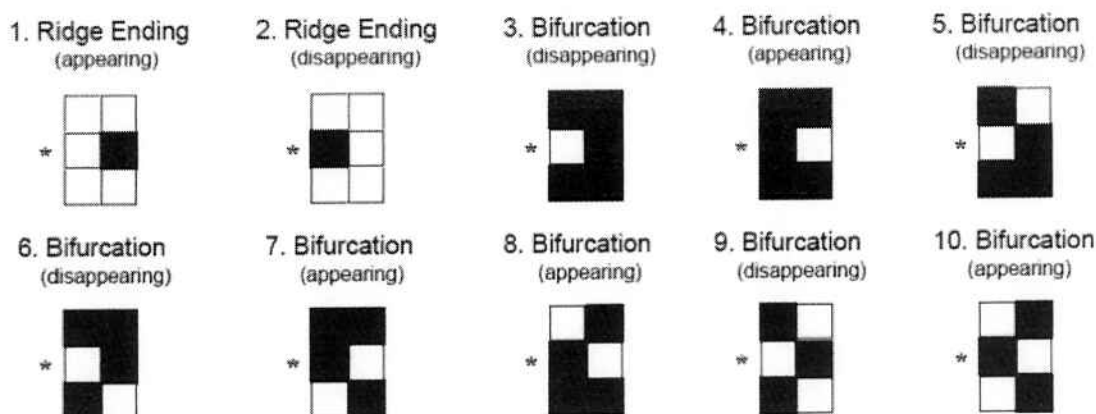


Figura 33 - Máscaras com padrões de minúcias do MINDTCT

Fonte: Implementation and Evaluation of NIST Biometric - Image Software for Fingerprint Recognition

Vale destacar que os asteriscos ao lado da linha de central de pixels de cada máscara significam que pode haver 1 a N daquelas linhas, isto é, cada máscara ilustrada acima corresponde a uma família de máscaras, capazes de identificar terminações e bifurcações de diferentes larguras. Desta forma, o uso do componente MINDTCT dispensa a necessidade de afinamento da imagem, etapa explicada anteriormente com finalidade didática, já que grande parte dos algoritmos de pré-processamento de digital envolve esse afinamento.

Após a identificação da posição da minúcia, inicia-se a análise da sua orientação, ela é calculada tendo como premissa que o ângulo aumenta no sentido anti-horário e a referência é o eixo (horizontal) X positivo, que corresponde à orientação de 0°.

A orientação de uma terminação é o ângulo entre a linha projetada a partir da terminação e o eixo X, enquanto a orientação da bifurcação é o ângulo entre a linha projetada no meio da bifurcação e o eixo X. As imagens a seguir ilustram a orientação para cada tipo de minúcia.

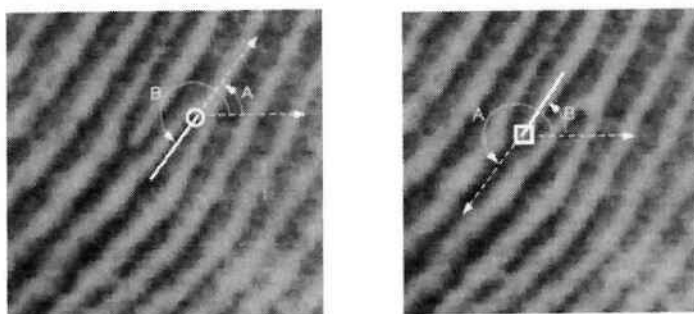


Figura 34 - Orientação de minúcias

Fonte: Autor

4.11.2 Eliminação de falsas minúcias

Sub-sistemas em que se aplica: 1, 2 e 3.

A eliminação de falsas minúcias ou validação de minúcias, também chamada de Pós-Processamento, tem como objetivo remover o máximo de falsas minúcias sem remover as minúcias verdadeiras.

As minúcias falsas podem ser dos seguintes tipos:

1. Minúcias adjacentes a outras
2. Minúcias próximas das bordas
3. Spike, break, bridge, hole e outras (ilustradas abaixo)

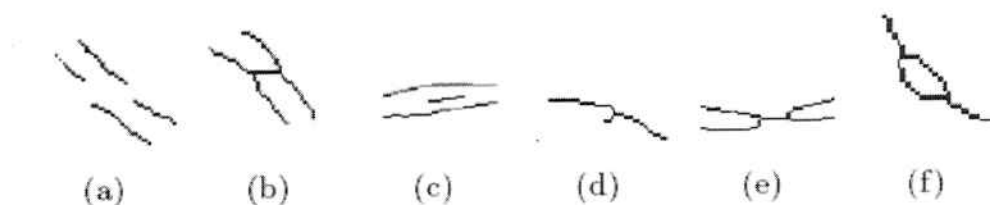


Figura 35 - Falsas minúcias. (a) cristas quebradas, (b) ponte, (c) crista curta, (d) crista curta, (e) crista curta, (f) buraco

Fonte: Autor

É necessário eliminar as falsas minúcias pois a grande maioria delas não está presente na impressão digital do usuário, mas sim é gerada artificialmente durante a etapa de afinamento, o que é ilustrado na Figura 36 - Geração de falsas minúcias no afinamento.

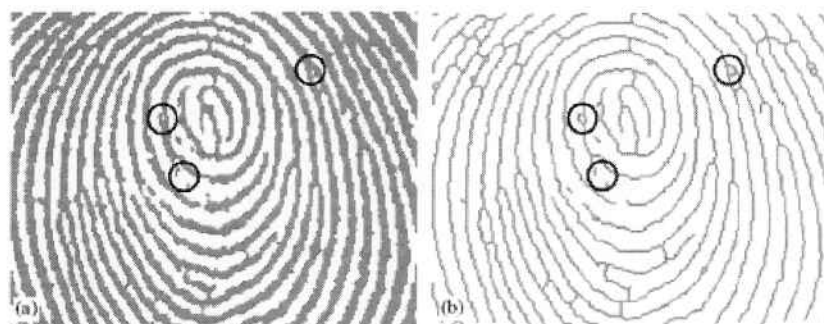


Figura 36 - Geração de falsas minúcias no afinamento

Há diferentes técnicas para identificar e eliminar falsas minúcias e neste documento explica-se uma delas.

Para cada candidata a minúcia, avalia-se sua vizinhança com uma matriz M de tamanho $W \times W$. W pode ser, por exemplo, igual a 20.

Para cada minúcia do tipo Terminação, são executados os seguintes passos durante a avaliação:

1. Centrar M na minúcia candidata, de forma que cada posição da matriz tenha um pixel correspondente da vizinhança.
2. Colocar -1 no espaço correspondente ao pixel central.
3. Preencher com 1 as posições de pixels que permitem conexão com o pixel central.
4. Dar uma volta completa na borda da máscara, sentido horário, calculando T_{01} = nº de transições de 0 para 1.
5. Se $T_{01} = 1$, a minúcia é verdadeira. Caso contrário, ela é removida do mapa.

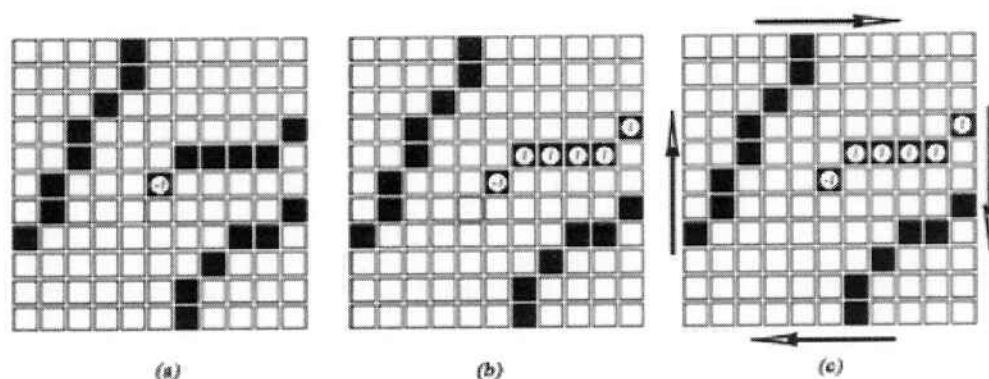


Figura 37 - Validação de um candidato a terminação. Passo-a-passo do algoritmo, com $W = 11$.

Para cada minúcia do tipo Bifurcação, são executados os seguintes passos durante a avaliação:

1. Centrar M na minúcia candidata, de forma que cada posição da matriz tenha um pixel correspondente da vizinhança.
2. Colocar -1 no espaço correspondente ao pixel central.
3. Dar uma volta completa, sentido horário, na vizinhança próxima, marcando os pixels conectados ao centro com 1, 2 e 3.
4. Rotular com 1 os pixels conectados ao pixel marcado com 1 no passo anterior. Ir rotulando até chegar numa borda ou em pixels com outra numeração. Repetir o processo de rotulação para os pixels de número 2 e 3.
5. Dar uma volta completa na borda da máscara, sentido horário, calculando T_{01} = nº de transições de 0 para 1, T_{02} = nº de transições de 0 para 2, T_{03} = nº de transições de 0 para 3.
6. Se $T_{01} = T_{02} = T_{03} = 1$, a minúcia é verdadeira. Caso contrário, ela é removida do mapa.

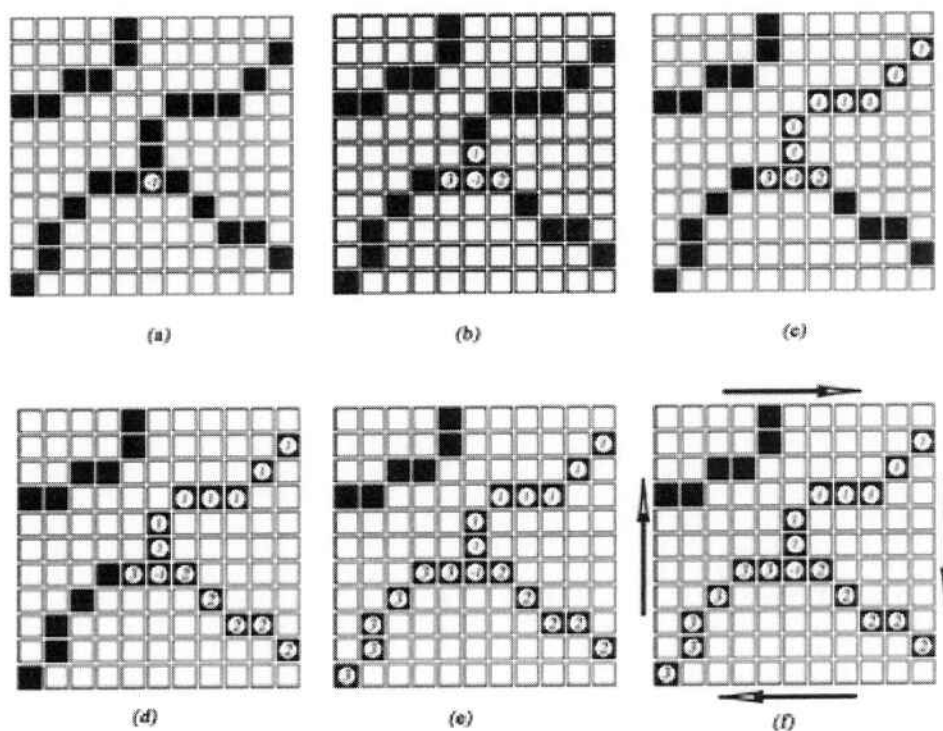


Figura 38 - Validação de um candidato a bifurcação. Passo-a-passo do algoritmo, com $W = 11$.

Exemplos de falsas minúcias, que são rejeitadas por esses testes:

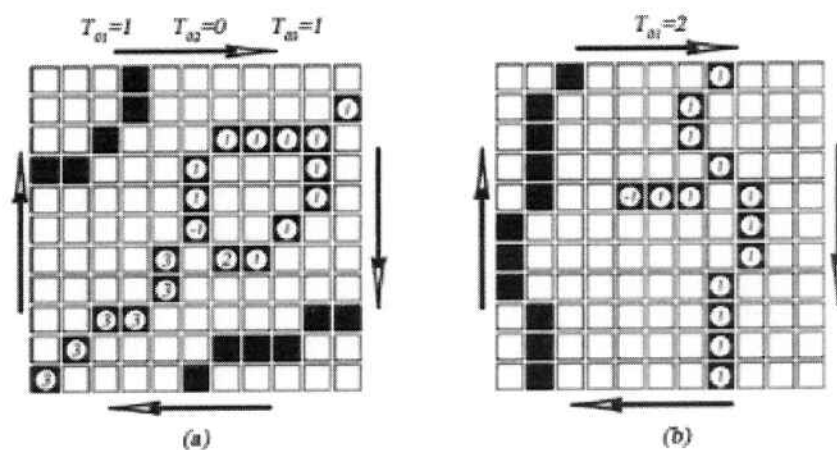


Figura 39 - Cancelamento de falsa minúcia. (a) buraco, (b) spike, com $W = 11$

Essa etapa de eliminação de falsas minúcias também é efetuada pelo MINDTCT, componente já mencionado na seção anterior.

4.12 Matching

A etapa de *Matching* consiste em comparar o *template* de minúcias obtido a partir da imagem de entrada com os *templates* armazenados no banco de dados. Cada comparação gera uma nota de similaridade que, se estiver acima de um limiar, resulta no fim da busca e na aceitação do usuário e, se estiver abaixo, implica na continuação da busca por um *template* similar no BD até que se compare o *template* de entrada com todos os registros no banco. Caso não haja similaridade entre o de entrada e nenhum *template* cadastrado, o usuário é rejeitado.

A nota de similaridade mínima funciona como uma regra para tomada de decisão e aqui podemos usar adaptatividade para que essa regra seja ajustada por aprendizado, conforme os erros e acertos do sistema.

A seguir será melhor explicada a geração da nota de similaridade, a decisão com base em regra pré-definida e a decisão com base em regra adaptativa.

4.12.1 Geração da nota de similaridade

Sub-sistemas em que se aplica: 1, 2 e 3.

Outro componente do *NBIS* recomendado pelo especialista Albert Nissimoff foi o Bozorth3. Ele é utilizado no projeto para realizar a comparação entre dois *templates* de minúcias, retornando uma nota de similaridade.

Para que a nota de similaridade não seja impactada por translações e rotações de um *template* em relação ao outro, eles são comparados em termos das características relativas entre suas minúcias, mais especificamente a distância e orientação relativas entre pares de minúcias. Assim, gera-se uma lista de pares de minúcias pertencentes a cada *template*, e a comparação é feita entre os pares de

um template e os do outro. Quanto mais pares de minúcias pertencerem aos dois, maior será a nota de similaridade.

A Figura 40 - Distância e orientação relativas entre minúcias ilustra a distância e orientação relativas entre pares de minúcias. Mesmo que haja rotação ou translação dos templates, essas informações relativas entre minúcias se mantêm iguais em ambos os templates.

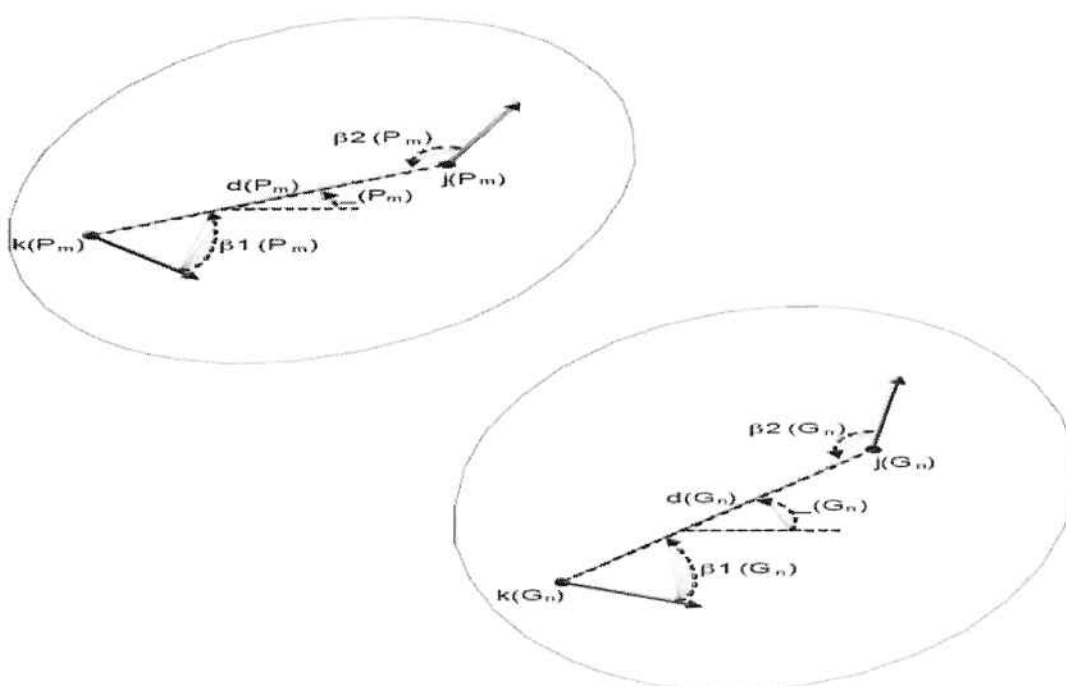


Figura 40 - Distância e orientação relativas entre minúcias

Fonte: MADDALA, S.; TANGELLAPALLY, S. R.: "Implementation and Evaluation of NIST Biometric Image Software for Fingerprint Recognition"

4.12.2 Decisão com base em nota de corte fixa

Sub-sistemas em que se aplica: 1 e 2.

Os *templates* extraídos e armazenados no banco de dados são comparados com o *template* de entrada e, conforme a nota de similaridade obtida, caso alguma nota seja superior ao valor fixado, o sub-sistema aceita o usuário.

Nos testes realizados, é possível avaliar o comportamento do sistema para diferentes notas de corte.

4.12.3 Decisão com base em nota de corte adaptativa

Sub-sistemas em que se aplica: 3.

Neste caso, a proposta é que a nota de corte deixe de ser fixa e passe a ser definida de forma adaptativa, a fim de otimizar métricas de desempenho.

O Sub-sistema 3 apresenta três modos de operação: padrão, tolerante e rigoroso. No modo padrão, que não é adaptativo, a nota de corte é fixa e inalterável. Já nos modos tolerante e rigoroso, o sub-sistema modifica a nota de corte de forma adaptativa e transparente ao usuário.

No caso do modo tolerante, comumente utilizado em aplicações em que a qualidade da impressão digital dos usuários é baixa e o nível de segurança não necessita ser muito elevado, a taxa de falsa rejeição (*FRR*) deverá ser reduzida.

Já no modo rigoroso, a taxa de falsa aceitação (*FAR*) deverá ser reduzida. Este modo visa atender aplicações com nível de segurança elevado ou com número de usuários bastante alto.

Para o funcionamento adequado dos dois modos de operação adaptativos, é necessário que o operador do sistema forneça *feedback*, indicando se cada indivíduo que tentou acessar é cadastrado ou não no sistema.

Com o *feedback* fornecido durante o treinamento, o sub-sistema é capaz de calcular suas taxas de falsa aceitação e falsa rejeição. Deste modo, de acordo com o modo de operação, o sub-sistema ajusta a nota de corte podendo reduzi-la ou aumentá-la, até chegar em um valor adequado para ser colocado em operação.

O objetivo é que a taxa de falsa aceitação do modo rigoroso oscile em torno de um valor estipulado, buscando estar sempre próxima a este valor. Para isso, o sub-sistema aumenta a nota de corte sempre que necessário para que a taxa de falsa aceitação fique no limiar do valor estipulado. Caso essa taxa apresente valor inferior ao número determinado, o sub-sistema reduz a nota de corte; caso contrário, ele aumenta a nota. O motivo da adoção desta regra é garantir que a nota de corte não atinja um valor que leve o sub-sistema à inoperabilidade, isto é, que seja necessário inserir a mesma impressão digital muitas vezes até que o usuário seja aceito.

Já o sub-sistema tolerante é análogo, porém visa minimizar a taxa de falsa rejeição (*FRR*), mantendo-a próxima ao valor determinado para a métrica. Inicialmente, a nota de corte é reduzida até que o sub-sistema atinja valor de *FRR* inferior ao limite estabelecido. Quando a taxa se torna inferior ao valor estipulado, o sub-sistema volta a aumentar a nota de corte; se *FRR* se torna superior ao limite, reduz-se a nota novamente. Dessa forma, a taxa de falsa aceitação tende a se manter próxima ao valor estipulado.

A figura a seguir ilustra o *tradeoff* entre as métricas *FRR* e *FAR*, e auxilia na compreensão do mecanismo de nota de corte adaptativa.

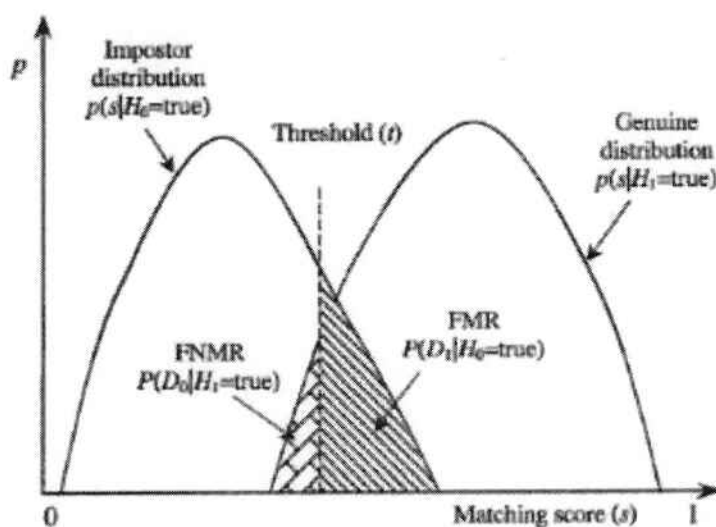


Figura 41 - Tradeoff entre métricas de desempenho

Fonte: MADDALA, S.; TANGELLAPALLY, S. R.: "Implementation and Evaluation of NIST Biometric Image Software for Fingerprint Recognition"

A nota de corte (*threshold*) é definida dinamicamente de forma a limitar a área sob uma das curvas ilustradas na Figura 41 - Tradeoff entre métricas de desempenho: *FAR* ou *FMR* (no caso do modo rigoroso) e *FRR* ou *FNMR* (no caso do modo tolerante).

Para que o cálculo dessas métricas seja possível, e consequentemente a lógica explicada, o sistema registra no banco de dados o número de falsas aceitações, falsas rejeições, aceitações verdadeiras e rejeições verdadeiras, atualizando esses números a cada tentativa de acesso no sistema. Evidentemente, é de extrema importância que o feedback enviado pelo operador do sistema seja preciso e confiável.

4.13 Interface Gráfica

As interfaces gráficas foram desenvolvidas com o Qt Creator 4.8 (<http://qt-project.org/>). Ele permite a codificação da interface em C++ e a integração com o projeto no Eclipse.

Para realizar a integração, foi necessário criar uma classe com os objetos a serem enviados para a interface gráfica, como as imagens tratadas em cada etapa do pré-processamento, os tempos de execução das etapas, a nota de similaridade, dentre outros itens.

Além disso, o projeto principal foi transformado em uma biblioteca que é adicionada ao Qt Creator. De acordo com a interação do operador com o sistema, determinadas funcionalidades do projeto são invocadas. Assim, algumas etapas do sistema são executadas e determinados objetos são enviados à interface gráfica, como imagens, tempos e nota de similaridade, entre outros itens.

A interface gráfica foi desenvolvida de modo que o operador do sistema possa visualizar de forma comparativa a distinção dos três sub-sistemas:

1. Sub-sistema sem pré-processamento da imagem e sem adaptatividade;
2. Sub-sistema com pré-processamento da imagem e sem adaptatividade;
3. Sub-sistema com pré-processamento da imagem e com adaptatividade.

Além disso, o operador pode visualizar a distinção entre os três modos de operação do sub-sistema adaptativo:

1. Tolerante;
2. Padrão;
3. Rigoroso.

A comparação entre os diferentes sub-sistemas e modos é facilitada pelo *layout* da interface: áreas diferentes, lado a lado, que exibem a imagem, o tempo de processamento, o número de minúcias extraídas, a decisão de aceitação ou rejeição, dentre outras informações, para cada sub-sistema e modo, permitindo a fácil visualização das diferenças.

Por fim, há uma interface específica para métricas, na qual é possível visualizar as taxas de falsa aceitação e falsa rejeição, além do tempo médio de *Matching*, tempo

de execução e outras métricas. Com isso, pretende-se identificar os sub-sistemas mais indicados para cada tipo de aplicação.

Nas seções a seguir, são apresentados as telas do sistema.

4.13.1 Telas de passo-a-passo do sistema

O passo-a-passo do sistema é composto por: seleção da imagem de entrada, etapas do pré-processamento de imagem, extração de minúcias e *matching*, conforme descritos nas sessões 4.9, 4.10, 4.11 e 4.12, respectivamente.

Para visualização clara e objetiva de cada etapa e seu comportamento para os diferentes sub-sistemas desenvolvidos, foram construídas as interfaces a seguir, que exibem não só a imagem modificada, mas também algumas métricas de desempenho relevantes para comparação entre os sub-sistemas.

Todas as telas de passo-a-passo do sistema possuem disponível a seleção de imagem para *upload*, além do rodapé com o *feedback* do operador do sistema. Após o clique no botão “Executar”, é possível selecionar outra imagem para execução somente após o envio do *feedback* da imagem que foi executada.

Nas etapas inexistentes em algum dos sub-sistemas, como ocorre na maioria dos passos para o Sub-sistema 1, a imagem exibida é a equivalente ao passo anterior.

Os botões de “Próxima Etapa” e “Etapa Anterior” permitem que o operador do sistema realize a navegação no passo-a-passo de forma cronológica, que é equivalente à ordem apresentada no menu lateral.

Já o menu lateral apresenta links para todas as etapas e permite que o operador possa ir diretamente para a etapa que lhe interessa.

Para melhor visualização de digitais com muitos detalhes, é possível clicar sobre qualquer uma das imagens e a interface abrirá uma nova janela com a imagem desejada em tamanho maior.

A figura a seguir exibe a etapa de janelamento, em que se adiciona uma borda branca à imagem.

FPoli - Pagina inicial Escola Politecnica da Universidade de Sao Paulo

Algoritmo

- >Janelamento
- >Normalização
- >Gabor
- >Binarização
- >Minúcias
- >Matching
- Métricas

Upload de Imagem fingerprints/original.tif Executar

Etapa: Janelamento Próxima Etapa

*Sem pré-processamento *Sem Adaptatividade		*Com pré-processamento *Sem Adaptatividade		*Com pré-processamento *Com Adaptatividade	
Tempo etapa		Tempo etapa		Tempo etapa	
Resultado	false	Resultado	false	Resultado	false
Tempo total	316,45 ms	Tempo total	310,63 ms	Tempo total	464,03 ms
Nº Minúcias	45	Nº Minúcias	73	Nº Minúcias	70
Qualidade min	43,95	Qualidade min	42,93	Qualidade min	49,98

Feedback ☐ Usuário não cadastrado ☐ Usuário cadastrado Enviar feedback

Figura 42 - Interface de passo-a-passo do sistema: Etapa de Janelamento

FPoli - Pagina inicial Escola Politecnica da Universidade de Sao Paulo

Algoritmo

- >Janelamento
- >Normalização
- >Gabor
- >Binarização
- >Minúcias
- >Matching
- Métricas

Upload de Imagem fingerprints/original.tif Executar

Etapa Anterior Etapa: Normalização Próxima Etapa

*Sem pré-processamento *Sem Adaptatividade		*Com pré-processamento *Sem Adaptatividade		*Com pré-processamento *Com Adaptatividade	
Tempo etapa		Tempo etapa	0,31 ms	Tempo etapa	0,35 ms
Resultado	false	Resultado	false	Resultado	false
Tempo total	316,45 ms	Tempo total	310,63 ms	Tempo total	464,03 ms
Nº Minúcias	45	Nº Minúcias	73	Nº Minúcias	70
Qualidade min	43,95	Qualidade min	42,93	Qualidade min	49,98

Feedback ☐ Usuário não cadastrado ☐ Usuário cadastrado Enviar feedback

Figura 43 - Interface de passo-a-passo do sistema: Etapa de Normalização

FPoli - Pagina inicial Escola Politécnica da Universidade de São Paulo

Algoritmo

- >Janelamento
- >Normalização
- >Gabor
- >Binarização
- >Minúcias
- >Matching
- Mettricas

Upload de Imagem  fingerprints/original.tif Executar

Etapa Anterior Etapa: Filtro de gabor Próxima Etapa

*Sem pré-processamento
*Sem Adaptatividade

*Com pré-processamento
*Sem Adaptatividade

*Com pré-processamento
*Com Adaptatividade

Tempo etapa	Resultado	Tempo etapa	Resultado	Tempo etapa	Tolerante	Padrão	Riguroso
Tempo etapa	false	Tempo etapa	false	Tempo etapa	104,08 ms	104,08 ms	104,08 ms
Tempo total	316,45 ms	Tempo total	310,63 ms	Tempo total	464,03 ms	459,20 ms	464,94 ms
Nº Minúcias	45	Nº Minúcias	73	Nº Minúcias	70	70	70
Qualidade min	43,95	Qualidade min	42,93	Qualidade min	49,98	49,98	49,98

Feedback ☐ Usuário não cadastrado ☐ Usuário cadastrado Enviar feedback

Figura 44 - Interface de passo-a-passo do sistema: Etapa de Filtro de Gabor

FPoli - Pagina inicial Escola Politécnica da Universidade de São Paulo

Algoritmo

- >Janelamento
- >Normalização
- >Gabor
- >Binarização
- >Minúcias
- >Matching
- Mettricas

Upload de Imagem  fingerprints/original.tif Executar

Etapa Anterior Etapa: Binarização Próxima Etapa

*Sem pré-processamento
*Sem Adaptatividade

*Com pré-processamento
*Sem Adaptatividade

*Com pré-processamento
*Com Adaptatividade

Tempo etapa	Resultado	Tempo etapa	Resultado	Tempo etapa	Tolerante	Padrão	Riguroso
Tempo etapa	false	Tempo etapa	5,07 ms	Tempo etapa	1,76 ms	1,76 ms	1,76 ms
Tempo total	316,45 ms	Tempo total	310,63 ms	Tempo total	464,03 ms	459,20 ms	464,94 ms
Nº Minúcias	45	Nº Minúcias	73	Nº Minúcias	70	70	70
Qualidade min	43,95	Qualidade min	42,93	Qualidade min	49,98	49,98	49,98

Feedback ☐ Usuário não cadastrado ☐ Usuário cadastrado Enviar feedback

Figura 45 - Interface de passo-a-passo do sistema: Etapa de Binarização

FPoli - Página inicial Escola Politécnica da Universidade de São Paulo

Algoritmo

- >Janelamento
- >Normalização
- >Gabor
- >Binarização
- >Minúcias
- >Matching
- Metricas

Upload de Imagem  fingerprints/original.tif Executar

Etapa Anterior Próxima Etapa

Etapa: Extração de minúcias

*Sem pré-processamento
*Sem Adaptatividade



*Com pré-processamento
*Sem Adaptatividade



*Com pré-processamento
*Com Adaptatividade



						Tolerante	Padrão	Rigoroso
Tempo etapa	67,68 ms	Tempo etapa	77,27 ms	Tempo etapa	79,05 ms	79,05 ms	79,05 ms	79,05 ms
Resultado	false	Resultado	false	Resultado	false	false	false	false
Tempo total	316,45 ms	Tempo total	310,63 ms	Tempo total	464,03 ms	459,20 ms	464,94 ms	464,94 ms
Nº Minúcias	45	Nº Minúcias	73	Nº Minúcias	70	70	70	70
Qualidade min	43,95	Qualidade min	42,93	Qualidade min	49,98	49,98	49,98	49,98

Feedback ☐ Usuário não cadastrado ☐ Usuário cadastrado Enviar feedback

Figura 46 - Interface de passo-a-passo do sistema: Etapa de Extração de Minúcias

FPoli - Página inicial Escola Politécnica da Universidade de São Paulo

Algoritmo

- >Janelamento
- >Normalização
- >Gabor
- >Binarização
- >Minúcias
- >Matching
- Metricas

Upload de Imagem  fingerprints/original.tif Executar

Etapa Anterior Próxima Etapa

Etapa: Matching

*Sem pré-processamento
*Sem Adaptatividade



*Com pré-processamento
*Sem Adaptatividade



*Com pré-processamento
*Com Adaptatividade



						Tolerante	Padrão	Rigoroso
Tempo etapa	248,77 ms	Tempo etapa	227,97 ms	Tempo etapa	269,12 ms	273,95 ms	274,86 ms	274,86 ms
Resultado	false	Resultado	false	Resultado	false	false	false	false
Tempo total	316,45 ms	Tempo total	310,63 ms	Tempo total	464,03 ms	459,20 ms	464,94 ms	464,94 ms
Nº Minúcias	45	Nº Minúcias	73	Nº Minúcias	70	70	70	70
Qualidade min	43,95	Qualidade min	42,93	Qualidade min	49,98	49,98	49,98	49,98

Feedback ☐ Usuário não cadastrado ☐ Usuário cadastrado Enviar feedback

Figura 47 - Interface de passo-a-passo do sistema: Etapa de Matching

4.13.2 Tela de métricas

A interface de métricas (Figura 48) apresenta todos os parâmetros calculados e relevantes para avaliação do desempenho dos sub-sistemas 1, 2 e 3 (todos os modos – padrão, tolerante e rigoroso).

As métricas do sistema são alteradas conforme ele é executado, ou seja, a cada acerto ou erro dos sub-sistemas, os valores das métricas são modificados.

FPoli - Métricas		Escola Politécnica da Universidade de São Paulo				
Algoritmo		Sistema 1	Sistema 2	Sistema 3 - Tolerante	Sistema 3 - Padrão	Sistema 3 - Rigoroso
Métricas	% falsa aceitação	14,28	33,33	14,89	8,95	4,76
	% falsa rejeição	28,31	47,05	15,18	27,67	29,09
	% verdadeira aceitação	85,71	66,66	85,10	91,04	95,23
	% verdadeira rejeição	71,68	52,94	84,81	72,32	70,90
	tempo médio	226,07 ms	221,51 ms	671,70 ms	653,84 ms	669,75 ms
Legendas: *Sistema 1: Sem pré-processamento / sem adaptatividade *Sistema 2: Com pré-processamento / sem adaptatividade *Sistema 3: Com pré-processamento / com adaptatividade						
						Voltar

Figura 48 - Interface de métricas dos sistemas

5 Implementação

Para implementação dos três sub-sistemas deste projeto, diversas ferramentas, linguagens e bibliotecas foram utilizadas. Elas são detalhadas a seguir.

5.1 C++

C++ é uma linguagem de programação multi-paradigma e de uso geral. A linguagem é considerada de médio nível, pois combina características de linguagens de alto e baixo níveis. Desde os anos 1990 é uma das linguagens comerciais mais populares, sendo bastante usada também na academia por seu grande desempenho e base de utilizadores.

A escolha da linguagem C++ para desenvolvimento do projeto foi feita pelos seguintes aspectos:

Por ser uma linguagem compilada, é facilmente utilizada não só por sistemas em servidores e desktops, como também por sistemas embarcados:

- Portabilidade do código fonte para outras plataformas, por exemplo, é possível adaptar o código para Windows, Mac e Linux;
- É uma linguagem orientada a objetos, o que facilita a reutilização de código, o encapsulamento e a manutenção;
- Fácil integração com a biblioteca OpenCV e o IDE QT-Creator.

5.2 Eclipse

Eclipse é um IDE para desenvolvimento Java, que também suporta várias outras linguagens a partir de plugins como C/C++, PHP, ColdFusion, Python, Scala e plataforma Android. Ele foi feito em Java e segue o modelo open source de desenvolvimento de software.

O projeto Eclipse foi iniciado na IBM que desenvolveu a primeira versão do produto e doou-o como software livre para a comunidade. O gasto inicial da IBM no produto foi de mais de 40 milhões de dólares. Hoje, trata-se do IDE Java mais utilizado no mundo. O software Eclipse tem a licença EPL (Eclipse Public License).

A escolha da linguagem Eclipse foi feita pelas seguintes características:

- Facilidade para compilação do projeto, já que o Makefile é gerado por ele;
- IDE possui recursos como navegação simples pelo código fonte (referências, declarações de variáveis, etc), refatoração e integração com o GDB (GNU debugger), o que facilita depurar o projeto.

5.3 QML

QML (*Qt Meta Language* ou *Qt Modeling Language*) é uma linguagem declarativa, baseada em JavaScript, para o desenvolvimento de aplicações centradas na interface gráfica.

Elementos QML são um sofisticado conjunto de blocos gráficos (ex: rectangle, image) e comportamentais (ex: state, transition, animation). Esses elementos podem ser combinados para fazer desde botões simples até programas completos habilitados para internet. Os elementos QML podem ser integrados e estendidos por componentes C++ por meio do Qt framework.

Vale destacar que o QML é uma linguagem cujo runtime é chamado Qt Declarative.

5.4 Qt Creator

Qt Creator é uma IDE (Ambiente Integrado de Desenvolvimento) multiplataforma desenhado para as necessidades dos desenvolvedores Qt. Seus componentes são descritos nas seções seguintes.

A adoção do Qt foi motivada pelos aspectos listados:

- Extensa documentação;
- Fácil aprendizagem, resultando em agilidade no desenvolvimento das interfaces;
- Integração simples com C++;
- Pelo fato de QML ser baseado em linguagem de script (Javascript) o código torna-se reduzido e de fácil entendimento.

5.4.1 Editor de código avançado

Fornece recursos que ajudam novos usuários Qt a iniciar o aprendizado e começar a desenvolver mais rápido e incrementam a produtividade dos desenvolvedores Qt experientes. Inclui:

- Editor de código com suporte a C++, QML e ECMAScript;
- Ferramentas de navegação rápida pelo código;
- Realçamento de sintaxe e auto-complemento de código;
- Suporte a refatoração de código;
- Ajuda sensível a contexto;
- Correspondência de parênteses e modo de seleção de parênteses.

5.4.2 Depurador visual

O depurador visual para C++ conhece a estrutura de muitas classes Qt, aumentando sua habilidade de exibir os dados do Qt claramente.

Adicionalmente, o Qt Creator exibe os dados brutos do GDB de modo limpo e conciso.

Com o depurador visual, é possível:

- Interromper a execução do programa;
- Percorrer o programa linha por linha ou instrução por instrução;
- Definir pontos de parada;
- Examinar o conteúdo da pilha de chamada, observadores e variáveis, locais e globais.

5.4.3 Construtores de GUI

Construtor integrado de GUI (Interface Gráfica com o Usuário) e formulários para projetos C++, que permite desenhar e construir rapidamente a GUI e os formulários, que são completamente funcionais e podem ser exibidos imediatamente para garantir que eles irão parecer e se comportar exatamente como pretendido.

5.5 SQLite

SQLite é uma biblioteca em linguagem C que implementa um banco de dados SQL embutido. Programas que usam a biblioteca SQLite podem ter acesso a banco de dados SQL sem executar um processo SGBD separado.

SQLite não é uma biblioteca cliente usada para conectar com um grande servidor de banco de dados, mas sim o próprio servidor. A biblioteca SQLite lê e escreve diretamente para e do arquivo do banco de dados no disco.

Os motivos que levaram à adoção do SQLite são:

- Dispensa necessidade de um processo servidor para o banco de dados, dessa forma o desempenho do sistema melhora, tornando-se mais veloz;
- Não necessita configurações, assim, o controle e segurança são estipulados pelo desenvolvedor do projeto, o que permite maior liberdade e praticidade no início do desenvolvimento.

Vale destacar que esta escolha foi feita para esse projeto acadêmico, porém o banco de dados mais adequado dependerá do tipo de aplicação. Evidentemente, instituições como FBI armazenam as digitais em BDs mais robustos, devido a maior necessidade de segurança.

Existem diversos servidores e sistemas gerenciadores de banco de dados no mercado, fornecidos por grandes empresas como Oracle e IBM. De acordo com os recursos financeiros disponíveis, é possível ter acesso a BDs cada vez mais robustos.

5.6 OpenCV

“*Open Source Computer Vision Library*”, trata-se de uma biblioteca de código aberto focada em visão computacional e aprendizado de máquina. A biblioteca possui mais de 2500 algoritmos otimizados que incluem um conjunto compreensível tanto de algoritmos clássicos quanto de “estado da arte”. Esses algoritmos podem ser utilizados para detectar e reconhecer rostos, identificar objetos, seguir movimentos

de olhos, além de fazer tratamento de imagens, como binarização, filtros, entre outras funcionalidades.

OpenCV tem interfaces para C++, C, Python, Java e MATLAB, e suporta Windows, Linux, Mac OS, iOS e Android. Além disso, possui mais de 47 mil pessoas na comunidade de usuários e um número estimado de *downloads* superior a 7 milhões.

Por ser uma biblioteca bastante ampla, com uma grande comunidade de desenvolvedores e por ter interface para várias linguagens, o OpenCV foi estudado profundamente para que o tratamento da imagem digital fosse feito por meio da utilização de funcionalidades desta biblioteca. Diversas dessas funcionalidades foram utilizadas, como Filtro de Gabor, normalização, binarização etc. Vale destacar que a maioria das funcionalidades foram adaptadas, visto que não atendiam completamente às necessidades deste projeto.

5.7 NIST Biometric Image Software (NBIS)

O NBIS é um sistema desenvolvido pelo governo americano para soluções biométricas. É desenvolvido em C e opera em sistemas Linux, MacOS e Windows NT.

Além disso, o projeto é *open source*; desta forma, é possível utilizar funcionalidades deste sistema para compor o projeto de formatura. Dentre as funcionalidades do NBIS, temos:

- NFIQ: algoritmo responsável pela análise e classificação da imagem de uma impressão digital de acordo com a qualidade da figura;
- PCASYS: rede neural responsável pela classificação da digital de acordo com o seu formato (arco, volta esquerda ou direita, espiral, entre outros);
- **MINDTCT**: detector de minúcias. Automaticamente identifica e armazena as minúcias do tipo bifurcação e terminação. Ele recebe uma imagem e gera um arquivo no formato . xyt contendo a coordenada X, a coordenada

Y, o ângulo de orientação Theta e uma nota de qualidade para cada uma das minúcias extraídas;

- **BOZORTH3**: algoritmo que compara dois *templates* de minúcias no formato .xyt e gera uma nota de similaridade;
- **NFSEG**: auxilia na remoção de espaços brancos da imagem, ao redor da impressão digital;
- **SIVV**: algoritmo responsável pela verificação espectral da imagem, dessa forma é possível identificar a presença ou não de uma digital na figura.

Neste projeto, foram utilizados apenas os componentes MINDTCT e BOZORTH3, pois além de serem recomendados por especialistas, também foram julgados os mais interessantes e úteis para a implementação deste projeto.

6 Testes e Resultados

A seguir é descrita a metodologia de teste utilizada como base para desenvolvimento da metodologia aplicada neste projeto, que também é detalhada em seguida.

Os resultados obtidos são avaliados para cada uma das métricas relevantes. A avaliação é feita em duas visões:

- Comparação entre os sub-sistemas 1, 2 e 3;
- Comparação entre os modos do sub-sistema 3.

6.1 Metodologia de testes da FVC

Para cada banco de dados:

- Cada amostra de um subgrupo A é comparada contra as amostras restantes de mesmo dedo para computar a taxa de falsa aceitação (*FRR – False Rejection Rate* ou *FNMR – False Non Matching Rate*). Se o *matching* X contra o Y é executado, o simétrico (Y contra X) não é executado para evitar correlação. O número de testes genuínos é de $((8*7)/2) * 100 = 2800$.
- A primeira amostra de cada dedo em um subgrupo A é comparada contra a primeira amostra dos demais dedos em A para computar a taxa de falsa aceitação (*FAR – False Acceptance Rate* ou *FMR – False Matching Rate*). Se o *matching* X contra o Y é executado, o simétrico (Y contra X) não é executado para evitar correlação. O número total de testes de falsa aceitação é $((100*99)/2) = 4950$

Para cada banco de dados, os seguintes indicadores de desempenho são computados:

- REJ_{ENROLL} (número de digitais rejeitadas durante o cadastramento)
- REJ_{NGRA} (número de digitais rejeitadas durante *matches* genuínos)
- REJ_{NIRA} (número de digitais rejeitadas durante *matches* impostores)
- Distribuição impostoras e genuínas
- Curvas $FMR(t)$ / $FNMR(t)$
- Curva ROC (t)
- EER (Equal Error Rate)
- EER^* (o valor que a EER assumiria se as falhas de *matching* fossem excluídas da computação da FMR e $FNMR$)
- $ZeroFMR$ (a menor $FNMR$ para a qual não acontecem falsos *matches*)
- $ZeroFNMR$ (a menor FMR para a qual não acontecem falsas rejeições)
- Tempo médio de cadastramento
- Tempo médio de verificação

Os seguintes indicadores de desempenho são reportados em valores gerais, sobre os quatro bancos de dados:

- EER média
- EER^* média
- REJ_{ENROLL} médio (número médio de digitais rejeitadas durante o cadastramento)
- REJ_{MATCH} médio (número médio de digitais rejeitadas durante *matches* genuínos e impostores)
- Tempo médio de cadastramento
- Tempo médio de verificação

Esse padrão para a realização dos testes e para reportar os resultados permite uma comparação objetiva e completa entre os algoritmos concorrentes da competição, para a determinação dos vencedores, bem como para avaliação comparativa da qualidade de outros algoritmos desenvolvidos no meio comercial ou acadêmico.

6.2 Metodologia de testes adotada

A metodologia de testes adotada neste projeto é uma adaptação simplificada da metodologia da *FVC*.

Para computar a taxa de falsa aceitação, é utilizado o DB1 de 2004 e não são realizadas todas as possíveis combinações entre as oito imagens de um mesmo dedo, mas sim, as sete últimas digitais são comparadas com a primeira (e com as demais cadastradas no sistema). O número de testes genuínos é de $(7 * 10) = 70$.

Para computar a taxa de falsa rejeição, é utilizado o DB1 em conjunto com o DB2 de 2004. São cadastradas as primeiras imagens de cada dedo do DB1 no banco de dados e elas são confrontadas com sete imagens de cada dedo, provenientes do DB2 (i.e. imagens de outros dedos). O número de testes de falsa aceitação é $(10 * (7 * 10)) = 700$.

As métricas de desempenho adotadas são:

- *FAR*
- *FRR*

A metodologia de testes foi simplificada, pois devido aos prazos deste projeto não houve tempo suficiente para serem realizados os testes completos.

6.3 Parâmetros adotados

Na realização dos testes, foram usadas diferentes combinações de parâmetros a fim de identificar o ponto ótimo de operação do sistema.

Os parâmetros variados na realização dos testes foram:

- **Nota de corte dos sub-sistemas 1, 2 e 3 modo padrão:** Valor da nota de corte para similaridade entre *templates*;
- **Nota de corte mínima do Sub-sistema 3 modo tolerante:** Valor mínimo que a nota de corte pode ter no modo tolerante do Sub-sistema 3. Ou seja, a nota de corte pode oscilar, porém nunca será inferior ao valor deste parâmetro;
- **Nota de corte mínima do Sub-sistema 3 modo rigoroso:** Valor mínimo da nota de corte do modo rigoroso do Sub-sistema 3. Similar ao tópico anterior;
- **Nota de corte inicial do Sub-sistema 3 modo tolerante:** Quando iniciado o sistema, ou seja, antes da realização de qualquer teste, a nota que o modo tolerante do Sub-sistema 3 apresenta;
- **Nota de corte inicial do Sub-sistema 3 modo rigoroso:** Idem ao tópico anterior, porém para o modo rigoroso do Sub-sistema 3;
- **Incremento ou decremento a cada passo da nota do Sub-sistema 3, modos tolerante e rigoroso:** A cada tentativa de acesso, de acordo com o *feedback* fornecido pelo operador, o Sub-sistema 3, modos tolerante e rigoroso, altera as notas de corte de cada um desses modos de operação. A alteração ocorre pelo incremento ou decremento de um valor. Este parâmetro indica qual o valor será adicionado ou subtraído da nota de corte atual de modo a compor a nova nota, buscando atender às especificações de cada modo;
- **Limite buscado pelo Sub-sistema 3 modo rigoroso para a taxa de falsa aceitação (FAR):** O modo rigoroso do Sub-sistema 3 visa reduzir o FAR, este parâmetro indica qual o valor buscado por esse modo, em porcentagem. Para atingir o valor estipulado, o sistema altera a nota de corte buscando que a taxa de falsa aceitação esteja sempre próxima a este valor;
- **Limite buscado pelo Sub-sistema 3 modo tolerante para a taxa de falsa rejeição (FRR):** O modo tolerante do Sub-sistema 3 busca diminuir o FRR até que atinja o valor buscado, indicado por este parâmetro. Alterando a nota de corte, o modo tolerante altera dinamicamente a taxa de falsa rejeição para que sempre esteja próxima ao valor estipulado por este parâmetro;
- **Quantidade mínima de testes para início da alteração da nota de corte dos modos tolerante e rigoroso do Sub-sistema 3:** Este parâmetro indica

quantos testes iniciais são necessários para que o sistema inicie o ajuste da nota de corte de acordo com o *feedback* do operador e o modo do sub-sistema.

Após testar diferentes combinações de parâmetros, os seguintes valores foram adotados:

- Nota de corte dos sub-sistemas 1, 2 e 3 modo padrão: 30
- Nota de corte mínima do Sub-sistema 3 modo tolerante: 14
- Nota de corte mínima do Sub-sistema 3 modo rigoroso: 30
- Nota de corte inicial do Sub-sistema 3 modo tolerante: 30
- Nota de corte inicial do Sub-sistema 3 modo rigoroso: 30
- Incremento ou decremento a cada passo da nota do Sub-sistema 3, modos tolerante e rigoroso: 1
- Limite buscado pelo Sub-sistema 3 modo rigoroso para a taxa de falsa aceitação (*FAR*): 5%
- Limite buscado pelo Sub-sistema 3 modo tolerante para a taxa de falsa rejeição (*FRR*): 5%
- Quantidade mínima de testes para início da alteração da nota de corte dos modos tolerante e rigoroso do Sub-sistema 3: 5

Na sequência, são apresentados os resultados dos testes em termos das métricas de desempenho.

6.3.1 Resultados dos testes

	Sub-sistema 1	Sub-sistema 2	Sub-sistema 3 - Tolerante	Sub-sistema 3 - Padrão	Sub-sistema 3 - Rigoroso
% falsa aceitação	14,28	33,33	14,89	8,95	4,76
% falsa rejeição	28,31	47,05	15,18	27,67	29,09

% verdadeira aceitação	85,71	66,66	85,10	91,04	95,23
% verdadeira rejeição	71,68	52,94	84,81	72,32	70,90
Tempo médio	226,07 ms	221,51 ms	671,70 ms	653,84 ms	669,75 ms

Tabela 1- Resultados dos testes

6.3.2 Conclusões sobre os resultados

Com os testes realizados, é possível observar que o sub-sistema 3, nos modos adaptativos – rigoroso e tolerante –, busca ajustar seu comportamento de modo a otimizar a métrica de interesse em cada caso. Observa-se que dentre todos os sub-sistemas, o tolerante obteve a taxa de falsa rejeição (*FRR*) mais próxima do valor estipulado (5%) e o rigoroso também para a taxa de falsa aceitação (*FAR*).

No caso do modo tolerante, quando a taxa de falsa rejeição está superior ao valor determinado, reduz-se a nota de corte. No momento em que a *FRR* do modo tolerante estiver inferior ao valor estipulado, o sub-sistema aumenta a nota de corte. O comportamento é similar no caso do modo rigoroso. Quando a taxa de falsa aceitação está superior ao valor estipulado, aumenta-se a nota de corte do modo rigoroso. Assim que o *FAR* torna-se inferior ao valor determinado, reduz-se a nota de corte. Dessa forma, conclui-se que a *FRR* (para o modo tolerante) e a *FAR* (para o modo rigoroso), a partir do momento que atingem seus valores-alvo, ficam sempre oscilando ao redor destes valores. Esse mecanismo foi utilizado para evitar que o sistema atingisse um modo inoperável, pois ao passo que a taxa de falsa aceitação está mais próxima de zero, maior é a taxa de falsa rejeição, e vice-versa.

É importante lembrar que o comportamento do sistema é dependente das imagens que recebe de entrada. Portanto, não há um valor ideal para a nota de corte, e isso é o que torna os modos adaptativos atraentes, pois os sub-sistemas modificam seu comportamento de acordo com as entradas e de acordo com os parâmetros indicados.

Portanto, em relação aos métodos adaptativos para ajuste da nota de corte e, consequentemente, modificação das taxas de falsa aceitação e falsa rejeição, é possível observar que os sub-sistemas são capazes de se adaptar às necessidades das aplicações, como observado nos testes.

Analizando agora os sub-sistemas 1 (sem pré-processamento e sem adaptatividade) e 2 (com pré-processamento e sem adaptatividade), observa-se que o desempenho do 1 é superior ao do 2. Isso ocorre, pois ao realizarmos o pré-processamento da imagem como um todo, sem fazer o janelamento, a qualidade desta acaba sendo prejudicada na maior parte dos casos, sobretudo devido à etapa de normalização, que aumenta o contraste da digital inteira em relação ao fundo branco, em vez de aumentar o contraste entre as cristas e vales da digital.

Já o pré-processamento feito por janelas apresenta um resultado satisfatório, como pode ser observado pelo desempenho do sub-sistema 3 padrão (com pré-processamento por janelas), que é superior ao desempenho do 1 e 2.

Por fim, vale ressaltar que o pré-processamento por janelas tem um custo significativo em termos de tempo. Observa-se que os tempos médios de execução dos sub-sistemas 3 são cerca de três vezes maiores que do 1 e 2. Contudo, ainda assim a ordem de grandeza desses tempos é de apenas 600 ms, o que é muito baixo do ponto de vista do usuário final.

Esses tempos foram calculados para um BD com cerca de 80 templates de digitais cadastrados. Quanto maior o número de templates, maior será o tempo e este pode vir a se tornar um problema para o usuário final, exigindo a busca de soluções como, por exemplo, a indexação do BD por tipo de digital.

7 Considerações finais

Com a conclusão deste projeto, muitos outros trabalhos complementares podem agregar a esta pesquisa.

É possível organizar os trabalhos complementares em dois grupos principais: expansão do algoritmo de reconhecimento de padrões para novos tipos de padrões; e expansão do uso de adaptatividade no algoritmo de reconhecimento de impressão digital. Cada um deles será comentado nas seções a seguir.

7.1 Possíveis trabalhos complementares relacionados a reconhecimento de padrões

Quanto ao reconhecimento de padrões, a partir de uma imagem são extraídos parâmetros relevantes que tornam identificável determinado objeto ou ser vivo. A aplicação escolhida foi o padrão de impressões digitais, mas também é possível expandir o reconhecimento para: outros padrões biométricos; padrões da natureza; padrões de doenças em exames médicos, dentre outros.

Para citar alguns exemplos, no caso de padrões biométricos, há diversas outras características que nos torna identificáveis, por exemplo o padrão da íris do olho, formato da mão, padrão de voz, formato da face e assinatura. É possível identificar os parâmetros relevantes de cada um desses itens e, a partir da extração deles, identificar os indivíduos.

No caso de padrões da natureza, um exemplo que pode ser citado é a identificação de espécies de abelhas a partir do desenho em suas asas. Outro exemplo são plantas, que podem ser identificadas a partir das ranhuras em suas folhas.

Também é possível identificar padrões para doenças, como câncer de pele, irregularidades no cérebro, dentre outras.

7.2 Possíveis trabalhos complementares relacionados à adaptatividade

Do ponto de vista de adaptatividade, é possível manter a mesma aplicação escolhida – autenticação por impressão digital – e englobar novos parâmetros ou incluir novas técnicas. Por exemplo, parâmetros internos da etapa de *matching*, relacionados ao cálculo da nota de similaridade, poderiam ser investigados e tornados adaptativos. O sistema definiria o ponto ótimo de operação desses parâmetros com base em treinamento, da mesma forma que faz para o parâmetro de nota de corte.

Outra possível aplicação de adaptatividade é o desenvolvimento de nota de corte personalizada para cada usuário do sistema. A sugestão é: a partir da retroalimentação do sistema realizada pelo operador, a aplicação avalia a qualidade da digital de cada usuário, podendo alterar a nota de corte de forma específica para cada pessoa cadastrada no sistema. Para que o sistema personalizasse a nota de corte, o algoritmo precisaria de algumas novas informações que atualmente não coleta.

Há duas possibilidades para que o sistema individualize o cálculo da nota de corte. A primeira opção é a inserção do feedback do operador indicando qual usuário tentou acessar – e não somente se é usuário ou não, como é feito atualmente.

A segunda possibilidade é: a cada rejeição indevida, o sistema armazena os usuários seguintes que tiveram seus acessos liberados. Dessa forma, a partir de uma coleção de registros, o sistema: (1) infere quais usuários apresentam maior dificuldade para se autenticar e reduz a nota de corte desses determinados indivíduos; ou (2) os usuários que são aceitos de forma indevida, por exemplo, Maria – que não é cadastrada no sistema – tenta acessar e é autorizada, identificada como

o usuário João, nesse caso, a nota de corte de João deve ser elevada para evitar que pessoas como Maria possam, indevidamente, acessar o sistema em seu lugar.

Ainda mais interessante e mais desafiador seria tornar o sistema capaz de identificar os próprios parâmetros de interesse, os quais ele deveria variar para melhorar o desempenho da autenticação, de forma que os pesquisadores não precisem pré-definir esses parâmetros.

7.3 Conclusão

Quanto mais amplamente forem aplicados conceitos e técnicas adaptativas ao algoritmo de reconhecimento de digitais, mais apto a reconhecer automaticamente diversos tipos de padrões ele se tornará. Isso reduz ou até elimina a necessidade de adaptação do código por parte dos pesquisadores a cada nova aplicação.

Por exemplo, sistemas de propósito geral, que possuem finalidades mais abrangentes, poderiam, a partir da análise do formato do objeto de entrada, identificar de que tipo ele é (uma impressão digital, um animal, uma planta, um órgão, etc.), dessa forma o sistema determina os parâmetros relevantes que devem ser extraídos do objeto a fim de classificá-lo.

Por fim, fica evidente que adaptatividade pode agregar muito valor ao reconhecimento de padrões e que há oportunidade para diversas outras pesquisas em ambos os temas e, sobretudo, combinando os dois temas, como foi o caso deste projeto.

8 Referências

- ALONSO-FERNANDEZ, F.; BIGUN, J.; FIERREZ, J.; FRONTHALER, H.; KOLLREIDER, K.; ORTEGA-GARCIA, J.: *Fingerprint Recognition*, 2009.
- BUENO, J. F.: Sistema automatizado de classificação de abelhas baseado em reconhecimento de padrões. Tese (Doutorado) – EPUSP, São Paulo, 2010.
- CASTRO, T. S.: “Identificação de Impressões Digitais Baseada na Extração de Minúcias,” Juiz de Fora, Minas Gerais, Brasil, 2008.
- CHEN, W.; GAO, Y.: “A Minutiae-based Fingerprint Matching Algorithm Using Phase Correlation,” 9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications, Glenelg, Australia, 2007.
- JAIN, A.; HONG, L.: “Classification of Fingerprint Images”, 1998.
- JIANG, X.; YAU, W. Y. Yau; SER, W.: “Fingerprint Image Processing for Automatic Verification,” IEEE 2nd International Conference on Information, Communication & Signal Processing, Singapore, 1999.
- JUNIOR, M. C.: “Biometris – Um Estudo Comparativo de Algoritmos de Reconhecimento de Impressões Digitais”.
- KULKARNI, V. K.; JAYADEVAN, R.; MALI, S. N.; ABHYANKAR, H. K.: “A New Ridge Orientation based Method of Computation for Feature Extraction from Fingerprint Images”, World Academy of Science, Engineering and Technology, Vol: 2. 2008
- LAVANYA, B. N.; RAJA, K. B.; VENUGOPAL, K. R.: “Fingerprint Verification based on Gabor Filter Enhancement,” International Journal of Computer Science and Information Security, vol. 6, pp. 138-144, 2009.

MADDALA, S.; TANGELLAPALLY, S. R.: "Implementation and Evaluation of NIST Biometric Image Software for Fingerprint Recognition". Master's Thesis, Blekinge Institute of Technology, Sweden, 2010.

NETO, J. J.: Adaptatividade: Generalização Conceitual. Memórias do WTA 2009, EPUSP, São Paulo, 2009. ISBN 978-85-86686-51-1

_____. Adaptive Technology and Its Applications. Encyclopedia of Artificial Intelligence. New York, v.1, p.: 37-44, 2009b.

_____. Adaptive Rule-Driven Devices: General Formulation and Case Study. Revista de Engenharia de Computação e Sistemas Digitais, São Paulo, v.1, n.1, p. 45-57, 2001.

_____. Solving complex problems with Adaptive Automata. Lecture Notes in Computer Science. S. Yu, A. Paun (Eds.): Implementation and Application of Automata 5th International Conference, CIAA 2000, Vol.2088, London, Canada, Springer-Verlag, pp.340, 2000.

_____. Contribuição à metodologia de construção de compiladores. 272p. Tese (Livre-Docência), EPUSP, São Paulo, 1993.

NETO, J. J., IWAI, M. K.: Adaptive Automata for Syntax Learning. In Anais da XXIV Conferência Latino-americana de Informática - CLEI 98, pg. 135–149, Quito, Equador, 1998.

NIXON, M. S.; AGUADO, A.S.: *Feature Extraction and Image Processing*. Newnes, 2002.

RAICEVIC, A. M.; POPOVIC, B. M.: "An Effective and Robust Fingerprint Enhancement by Adaptive Filtering in Frequency Domain", Sérvia, 2009.

THAI, R.: "Fingerprint Image Enhancement and Minutiae Extraction," University of Western Australia, Perth, Western Australia, Australia, 2003.

THEODORIDIS, S.; KOUTOUMBAS, K.: *Pattern Recognition* 3ª Edição. Academic Press, 2006.