Universidade do Estado do Rio de Janeiro
Centro de Tecnologia e Ciências
Faculdade de Engenharia

Rafael Soares Wyant

Implementação de verificações biométricas processadas em cartões inteligentes a multi-aplicações

Rio de Janeiro
2013

Rafael Soares Wyant

Implementação de verificações biométricas processadas em cartões inteligentes a multi-aplicações

Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-Graduação em Engenharia Eletrônica, da Universidade do Estado do Rio de Janeiro. Área de concentração: Sistemas Inteligentes e Automação.

Orientadora: Prof.ª Dr.ª Nadia Nedjah
Orientadora: Prof.ª Dr.ª Luiza de Macedo Mourelle

Rio de Janeiro
2013

CATALOGAÇÃO NA FONTE
UERJ / REDE SIRIUS / BIBLIOTECA CTC / B

W143 Wyant, Rafael Soares
Implementação de verificações biométricas processadas em cartões inteligentes a multi-aplicações/Rafael Soares Wyant. – 2013.
126 f. : il.

Orientadora: Nadia Nedjah.
Orientadora: Luiza de Macedo Mourelle.
Dissertação (Mestrado) – Universidade do Estado do Rio de Janeiro, Faculdade de Engenharia.

1. Engenharia Eletrônica - Dissertações. 2. Sistemas inteligentes - Dissertações. 3. Cartão inteligente. 4. Biometria. I. Nedjah, Nadia. II. Mourelle, Luiza de Macedo. III. Universidade do Estado do Rio de Janeiro. III. Título.

CDU 004.272.2

Assinatura

Data

Rafael Soares Wyant

Implementação de verificações biométricas processadas em cartões inteligentes a multi-aplicações

Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-Graduação em Engenharia Eletrônica, da Universidade do Estado do Rio de Janeiro. Área de concentração: Sistemas Inteligentes e Automação.

Aprovado em:

Banca Examinadora:

Prof.ª Dr.ª Nadia Nedjah (Orientadora)
Faculdade de Engenharia - UERJ

Prof.ª Dr.ª Luiza de Macedo Mourelle (Orientadora)
Faculdade de Engenharia - UERJ

Prof. Dr. Adolfo Bauchspiess
Faculdade de Tecnologia - UnB

Prof. Dr. Leandro Augusto Frata Fernandes
Instituto de Computação - UFF

Rio de Janeiro
2013

RESUMO

WYANT, Rafael Soares *Implementação de Verificações biométricas processadas em cartões inteligentes a multi-aplicações.* 2013. 126f. Dissertação (Mestrado em Engenharia Eletrônica) – Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2013.

As biometrias vêm sendo utilizadas como solução para sistemas de controle de acesso à diversos sistemas há anos, mas o simples uso da biometria não pode ser considerado como solução final e perfeita. Muitos riscos existem e não devem ser ignorados. A maioria dos problemas está relacionada ao caminho de transmissão entre o local onde os usuários requerem seus acessos e os servidores onde são guardados os dados biométricos capturados em seu cadastro. Vários tipos de ataques podem ser efetuados por impostores que desejam usar o sistema indevidamente. Além dos aspectos técnicos, existe o aspecto social. É crescente a preocupação do usuário tanto com o armazenamento quanto o uso indevido de suas biometrias, pois é um identificador único e, por ser invariável no tempo, pode ser perdido para sempre caso seja comprometido. O fato de que várias empresas com seus diferentes servidores guardarem as biometrias está causando incomodo aos usuários pois as torna mais suscetíveis à ataques. Nesta dissertação, o uso de cartões inteligentes adotado como possível solução para os problemas supracitados. Os cartões inteligentes preparados para multi-aplicações são usados para realizar as comparações biométricas internamente. Dessa forma, não seria mais necessário utilizar diversos servidores pois as características biométricas estarão sempre em um único cartão em posse do dono. Foram desenvolvidas e implementadas três diferentes algoritmos de identificação biométrica utilizando diferentes características: impressão digital, impressão da palma da mão e íris. Considerando a memória utilizada, tempo médio de execução e acurácia, a biometria da impressão da palma da mão obteve os melhores resultados, alcançando taxas de erro mínimas e tempos de execução inferiores à meio segundo.

Palavras-chave: Biometria. Cartões inteligentes. Minúcias. PalmCode. IrisCode

ABSTRACT

The biometrics have been used as a solution for access control systems for many years, but the simple use of biometrics can not be considered as final and perfect solution. There are many risks that should not be ignored. Most problems are related to the transmission path between the system where the users require access and the servers where the captured biometric data is stored. Various types of attacks can be made by impostors who want to use the system improperly. Besides the technical aspects, there is the social aspect. There is a growing concern of users about both data storage and the misuse of their biometrics, which is an unique identifier and, being invariant in time , may be lost forever if compromised. The fact that several companies keep their biometric data in different servers is causing discomfort to users because it makes their biometric data more susceptible to attacks. In this thesis, the use of smart cards is adopted as a possible solution to the above problems. Smart cards prepared for multi-applications are used to perform biometric comparisons internally. Thus, it would not be necessary to use different servers because biometric features will always be on a single card in the possession of the owner. It was developed and implemented three different algorithms using different biometric identification characteristics: fingerprint, palmprint and iris. Considering the used memory, average execution time and accuracy, palm print biometrics obtained the best results, achieving minimum error rates and processing time lower than half a second.

Keywords: Biometrics. Smart cards. Minutiae. PalmCode. IrisCode

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ALGORITMOS

SUMÁRIO

INTRODUÇÃO

A CORRETA identificação de uma pessoa por outra pessoa é uma tarefa instintiva. Uma pessoa pode naturalmente reconhecer outra por características físicas como forma do rosto, cor do cabelo, forma e cor dos olhos e altura, entre tantas outras. Além das características físicas, uma pessoa pode ser reconhecida por características comportamentais, tais como forma de falar, vocabulário usado e forma de andar, entre outras. Os sistemas biométricos são desenvolvidos para utilizar essas características para o mesmo fim, ou seja, reconhecer pessoas. Os sistemas biométricos utilizam computadores para executar a tarefa do reconhecimento, logo, esse reconhecimento deve ser feito através de medições das características avaliadas.

As biometrias são estudos de determinadas características humanas físicas ou comportamentais que sejam capazes de distinguir duas pessoas. Existem algumas propriedades que devem estar presentes nas características para que elas possam ser usadas como uma biometria. Todas as pessoas ou pelo menos a grande maioria devem possuir essa característica. Essa propriedade é denominada *universalidade*. Duas pessoas devem apresentar diferenças nessa característica. Essa propriedade é denominada *distinção*. A característica estudada deve ser invariante em relação ao tempo. Essa propriedade é denominada *permanência*. Finalmente, deve ser possível medir essa característica. Essa propriedade é denominada *coletabilidade* (JAIN; ROSS; PRABHAKAR, 2004). Atualmente, existem muitas características ou órgãos utilizados em biometrias, alguns exemplos são: DNA, orelha, face, veias da mão, impressão digital, geometria da mão, íris, impressão da palma da mão, retina, voz, assinatura e forma de andar, sendo as duas últimas características comportamentais.

A principal aplicação das biometrias é relacionada ao controle de acesso, ou seja, através do uso da biometria, uma pessoa pode ter seu acesso permitido ou negado. Esse acesso pode estar definido de várias formas, ou seja, pode estar relacionado à transposição física de uma porta ou a um saque em um caixa automático. Qualquer sistema que

necessite a correta identificação ou autenticação de um indivíduo pode fazer uso das biometrias. As biometrias, na maioria dos casos, possui vantagens quando comparadas à outros tipos de certificação de identidade pois somente elas podem realmente garantir a autenticidade do requerente.

Os sistemas bancários, por exemplo, normalmente recorrem ao uso de senhas de números e códigos de letras além da inserção de um cartão para garantir o acesso de um cliente às informações de sua conta. O uso de senhas e porte do cartão podem ser burlados por um terceiro com ou sem a permissão do proprietário. A senha pode até mesmo ser vista e decorada e o cartão roubado. Outro tipo de sistema que requer controle de acesso é o predial. Normalmente requerem apenas a aproximação de um cartão que pode ser simplesmente passado para um terceiro ou roubado por outra pessoa para que o erro de identificação ocorra. Um caso mais atual é o de compras por internet que também se restringem ao uso de senhas e informações pessoais. Em todos os casos citados, a implementação de um sistema biométrico poderia elevar consideravelmente a segurança contra usos indevidos. Apesar da utilização da biometria ser uma solução que visa aumentar a segurança, o risco de fraude não pode ser ignorado. Muitos desenvolvedores fazem isso acreditando que o uso da biometria é a solução final e perfeita para todos os problema de identificação (HACHEZ; QUISQUATER; KOEUNE, 2000).

As possibilidades de fraudes em um sistema de autenticação biométrica são muitas e algumas de difícil resolução. Alguns possíveis pontos de ataque desse tipo de sistema são enumerados em (AO; REN; TANG, 2008). Note que a maioria dos problemas não são exclusivos de um sistema biométrico.

- Identidade falsa: O impostor se apresenta com documentos falsos de outra pessoa e faz o cadastro das suas biometrias.

- Falsas características biológicas: O impostor frauda o sistema imitando a característica usada na biometria. Um exemplo seria uma cópia de uma impressão digital feita de borracha.

- Ataque intermediário: Durante a transferência dos dados extraídos da característica para o seu cadastro, o impostor modifica os dados para que sejam cadastrados os seus dados.

- Ataque ao banco de dados: O impostor modifica ou copia os dados contidos no banco de dados.

- Ataque por repetição: O impostor bloqueia a informação autêntica durante a transmissão e os reproduz quando deseja garantir seu acesso.

Além dos problemas com segurança, existem os problemas de aceitação por parte dos usuários. A biometria tem se difundido cada vez mais rapidamente no mundo e as pessoas estão começando a pensar na própria segurança quando cadastram indiscriminadamente suas biometrias em diversas instituições. Afinal, as biometrias estariam presentes em vários bancos de dados e mais suscetíveis à ataques. Dessa forma, a biometria, que em teoria é única e invariável no tempo, estaria para sempre comprometida. Recentemente, o assunto de roubo da biometria foi abordado pelo imprensa na divulgação de um aparelho telefônico que usa a impressão digital para desbloqueá-lo. Foram levantados alguns problemas como a fraude por imitação, utilizando luvas de borracha e outros como vírus e outros softwares infecciosos que poderiam captar informação dos sensores e enviá-la à facções criminosas (STEINBERG, 2013). Isso ilustra a preocupação existente em torno da biometria como representação da identidade de uma pessoa.

Para resolver os problemas mencionados, várias abordagens são possíveis como o uso de um esquema rigoroso de criptografia para evitar os ataques intermediários ou a fusão de mais de uma biometria para dificultar a fraude por imitação. Uma outra abordagem para evitar os problemas relacionados à transmissão e armazenamento em servidores seria o uso de cartões inteligentes para armazenar os dados da biometria. Dessa forma não haveriam longas transmissões dos dados sigilosos tão pouco a possibilidade de invasão do servidor para modificar ou roubar as informações biométricas. Para aumentar ainda mais a segurança, um cartão inteligente dotado de meios de processamento deve, ele mesmo, realizar a comparação biométrica evitando assim a transmissão da biometria armazenada. Ainda assim, alguns problemas de transmissão como o ataque intermediário ainda poderiam ocorrer mas certamente o caminho da transmissão seria menor e, além disso, os cartões também são preparados para transmissões criptografadas.

Alguns cartões inteligentes são capazes de armazenar e processar várias aplicações. Dessa forma, um único cartão poderia ser usado por diversas empresas e instituições. Essa capacidade traz uma possível resposta à crescente preocupação dos usuários em relação à segurança de sua identidade pois o processo de cadastro poderia ser feito totalmente

offline e os dados biométricos estariam sempre em posse do proprietário, armazenados no cartão. Dessa forma, a biometria cadastrada não ficaria em diversos servidores mas sim em apenas um cartão que atenderia às necessidades de diversas empresas.

Nesta dissertação é estudada a abordagem que faz uso de cartões inteligentes em conjunto às biometrias com o objetivo principal de aumentar a segurança de sistemas de controle de acesso. O objetivo é avaliar a possibilidade de usar um cartão inteligente multi-aplicação para realizar comparações biométricas. Para isso, foram implementados três tipos diferentes de comparações biométricas (impressão digital, impressão da palma da mão e íris) e, em seguida, seus desempenhos foram analisados e comparados. A organização desta dissertação e um resumo dos capítulos são apresentados a seguir.

Inicialmente, o Capítulo 1 apresenta uma introdução teórica sobre cartões inteligentes e biometrias, que são as principais tecnologias utilizadas no projeto desta dissertação. É apresentada a plataforma escolhida para desenvolvimento em cartões e suas principais características. Também são apresentadas as principais características das biometrias assim como uma comparação entre as mais usadas atualmente. Maiores detalhes sobre as biometrias escolhidas para implementação também são introduzidos.

O Capítulo 2 relata diversos trabalhos relacionados às três biometrias estudadas. Impressão digital, impressão da palma da mão e íris. As ideias principais atrás dos trabalhos relacionados à essas biometrias introduzidas. Esse capítulo apresenta os principais algoritmos estudados antes da escolha do algoritmo a ser implementado.

Os Capítulos 3, 4 e 5 mostram os detalhes do trabalho realizado para as biometrias da impressão digital, impressão da palma da mão e íris, respectivamente. Para cada uma das biometrias, é explicada a forma de extração, o algoritmo de comparação implementado, o banco de dados utilizado, os detalhes de implementação do algoritmo escolhido e os resultados alcançados. Cada biometria possui suas peculiaridades no tipo de extração das principais características. O algoritmo é explicado com maiores detalhes. O banco de dados escolhido assim como alguns exemplos de imagens são mostrados. Alguns detalhes da implementação do algoritmo utilizando as capacidades restritas do cartão são apresentados. Para obter o resultado final, foram feitos diversos testes utilizando diferentes parametrizações. Em cada teste foram realizadas milhares de comparações biométricas.

O Capítulo 6 traz uma comparação dos resultados entre as três biometrias estudadas. As comparações são feitas considerando a memória utilizada, o tempo de execução

médio de cada teste e a acurácia alcançada. É avaliada a possibilidade de implementação real utilizando essas biometrias.

Finalmente, o Capítulo 7 apresentada as conclusões a cerca das implementações realizadas assim como algumas orientações para trabalhos futuros a serem realizadas na área de biometrias em conjunto com cartões inteligentes.

Capítulo 1

SMART CARDS E BIOMETRIAS

BIOMETRIAS e *Smart Cards* já são, individualmente, ótimas ferramentas de segurança e o uso das duas em conjunto tende a formar uma nova ferramenta ainda mais segura. A Seção 1.1 traz as principais características dos *Smart Cards*, o seu uso e as plataformas de desenvolvimento. A Seção 1.2 explica a importância das aplicações biométricas, os vários tipos existentes. Alguns conceitos sobre a aplicação de biometrias usando cartões inteligentes são apresentados na Seção 1.3.

1.1 Cartões inteligentes

Cartões inteligentes ou *Smart Cards* são cartões feitos de plástico que possuem *chips* embutidos. Alguns modelos são capazes apenas de guardar dados enquanto outros também conseguem processá-los. Podem se conectar a uma leitora diretamente por contato físico ou sem contato usando radio frequência. Existem também os cartões híbridos que possuem as duas interfaces. As características dos cartões obedecem aos padrões internacionais (ISO/IEC 7816 e ISO/IEC 14443). Estão em constante desenvolvimento e podem carregar um volume considerável de dados.

Os cartões inteligentes são confeccionados por diversos fabricantes e estão disponíveis nas mais diversas configurações de processamento e memória. Os processadores podem ser de 8, 16 ou 32 *bits*, operando em frequências variando entre 1 e 7,5 MHz. A memória pode chegar a 512Kb, sendo esta dividida em ROM ou memória *flash*, EEPROM e RAM. Atualmente, trata-se de uma tecnologia comum e altamente difundida, usada em diversas aplicações, tais como:

- Controle de acesso: Usados como chave de identificação em empresas, prédios, passaportes e sistemas online;

- Sistemas de pagamento: Cartões de metrô e ônibus;
- Sistemas de telecomunicações: Cartões SIM.

Mais de 5 bilhões de cartões são fabricados anualmente (ALLIANCE, 2012). Todos os telefones que utilizam a tecnologia GSM possuem um cartão inteligente (SIM *card*). O sistema bancário mundial está migrando de fitas magnéticas para cartões inteligentes, objetivando o aumento da segurança. Os cartões mais recentes são capazes de suportar várias aplicações podendo estas terem acesso umas as outras ou não. Apesar de cartões multi-aplicação ainda não serem muito utilizados, possuem um grande potencial em aplicações tais como a unificação de documentos de identificação de um país (Registro de Cidadãos, Habilitação de motorista, Registro de trabalhador, etc.), o uso de apenas um cartão para várias empresas de venda, um cartão único de acesso para diversos edifícios, ou até mesmo ambas funções.

Existem diversas especificações para o desenvolvimento de aplicações a serem executadas nos cartões inteligentes como o Java Card, .NET, Multos, entre outras. O Java Card foi escolhido por razões a serem abordadas na Seção 1.1.2. A base para a execução da máquina virtual do Java Card é o *Global Plataform*. Na Seção 1.1.1 serão vistas suas principais características.

1.1.1 Plataforma Global

O *Global Platform* possui uma especificação complexa porém muito importante no processo de desenvolvimento de aplicações Java Card assim como de outras plataformas. Serão vistas as principais características necessárias para a implementação do projeto desta dissertação. A Seção 1.1.1.1 introduz o conceito do gerenciador do cartão (*Card Manager*) e a Seção 1.1.1.2 aborda o método de identificação das diferentes aplicações que um cartão pode armazenar simultaneamente.

1.1.1.1 Gerenciador do cartão

O *Card Manager* é o processo mais importante do *Global Platform*. Todos os cartões que implementam a especificação *Global Platform* possuem um. Este gerenciador implementa o domínio de segurança, que tem a função de gerenciar a autenticação entre *Host* e cartão através de uma conexão segura, o ciclo de vida do cartão e aplicativos, carga, instalação e exclusão de aplicativos (Applets para o caso de Java Cards) e as chaves do algoritmo

criptográfico usado para operações em ambiente seguro. O Card Manager pode ser acessado utilizando-se o comando *select* (referente a especificação de *Smart Card* - ISO 7816-4) junto ao identificador da aplicação (*AID - Application Identifier*). Após selecionado existe uma série que funções que podem ser chamadas como a transferência ou instalação de uma aplicação, solicitação de status do cartão e retirada de uma aplicação entre outros.

1.1.1.2 Identificador de aplicação

Todos os aplicativos para *Smart Card* possuem um identificador de aplicação. o Card Manager do cartão de Crédito/Débito com Chip, o SIM Card GSM para celulares ou de qualquer outro tipo de cartão usa esse identificador para diferenciar as aplicações. Basicamente, o identificador é utilizado para selecionar a aplicação a ser executada dentro do cartão. Não podem existir dois identificadores iguais pois o *Card Manager* poderia se comunicar com o aplicativo incorreto. o AID segue os Padrões ISO 7816 e tem o seguinte formato [RID + PIX]:

- RID (*Registered Provider Identifier*), consiste de obrigatoriamente 5 bytes que identificam o fabricante/desenvolvedor da aplicação.
- PIX (*Proprietary Application Identifier Extension*), consiste no identificador de Aplicativo e é definido pelo fabricante/desenvolvedor do aplicativo.

A escolha e controle desses identificadores devem ser feitos por um mesmo órgão afim de evitar possíveis coincidências. Nas aplicações desenvolvidas nesta dissertação não há necessidade deste controle pois os aplicativos são desenvolvidos em cartões de teste e não possuem nenhum fim comercial.

1.1.2 Java Card

Java Card oferece uma plataforma de desenvolvimento bem difundida e utilizada em diversos setores tais como telefonia móvel com os cartões SIM, industria financeira e sistemas de pagamento (principalmente após o surgimento dos cartões sem contato), identificação, segurança de acesso, transporte público, TV paga e controle de acesso privado entre outros. A plataforma possui ferramentas de desenvolvimento integradas com simuladores e um grande número de ferramentas para auxiliar no desenvolvimento e execução do projeto em desenvolvimento. Seguem alguns dos benefícios que a plataforma proporciona:

- Interoperabilidade: A aplicação desenvolvida, denominada *Applet*, pode ser executada em qualquer outro Java Card de mesma versão independente do fabricante ou das características específicas do hardware (chip) no qual está sendo executada;
- Segurança: Possui execução segura herdada da linguagem Java;
- Capacidade de multi-aplicação: Em um único cartão inteligente, é possível coexistirem vários aplicativos de forma segura;
- Natureza dinâmica: Novas aplicações podem ser instaladas com segurança após a fabricação de acordo com as necessidades dos usuários;
- Compatibilidade com os padrões existentes: A Interface de Programação de Aplicações (*API - Application Programming Interface*) é compatível com os padrões internacionais, como o ISO 7816 entre outros.

O Java Card opera sobre a camada *Global Plataform*, como abordado na Seção 1.1.1, é um tipo de gerente de aplicações comuns a todas as plataformas. Todas as trocas de mensagens devem necessariamente passar por essa camada, mas a execução se dá no núcleo do Java Card. A Figura 1 mostra uma ilustração da arquitetura Java Card. Note que a camada *Global Plataform* é referenciada como sistema nativo pois o Java Card foi feito para ser independente do sistema de base.

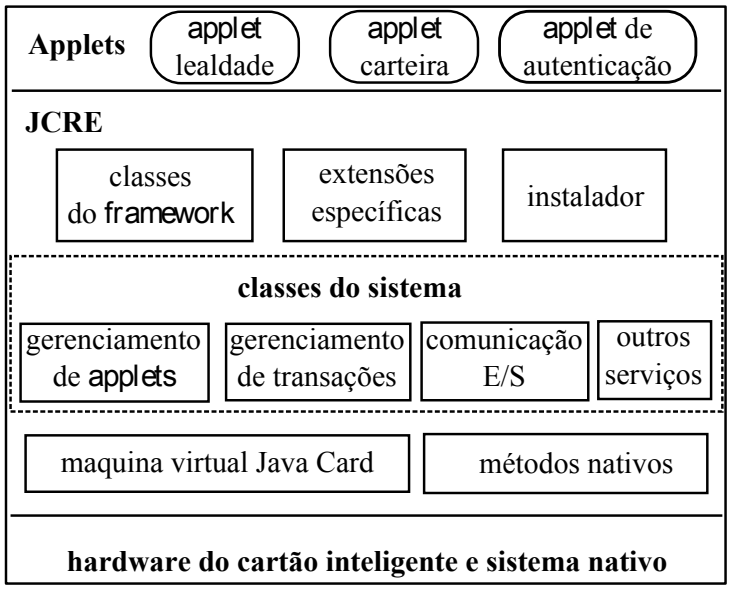


Figura 1: Arquitetura Java Card

Para o projeto desta dissertação é uma necessidade a existência de uma natureza dinâmica e a capacidade de multi-aplicação além da conveniente característica de interoperabilidade.

Por ser executado em um dispositivo mais simples, existem várias diferenças entre o Java convencional e Java Card. As características que continuaram presentes no Java Card:

- Tipos primitivos de dados menores: *Boolean, byte, short*;
- Vetores a uma dimensão;
- Pacotes, classes, interfaces e exceções;
- Características da orientação a objetos.

Tendo em vista o baixo poder de processamento, uma RAM de tamanho reduzido, arquitetura mais simples do dispositivos entre outras razões, algumas das funcionalidades mais elaboradas do Java convencional não estão presentes no Java Card. As principais destas são listadas a seguir:

- Tipos pimitivos maiores: *int, long, double, float*;
- Caracteres e *strings*;
- Vetores multidimensionais;
- Carregamento dinâmico de classes;
- Coletor de lixo (*garbage collection*);
- Threads;
- Serialização de objetos;
- Clonagem de objetos.

Essa série de restrições existentes no Java Card torna o desenvolvimento de aplicações mais desafiador pois o torna mais próximo de uma linguagem de programação de nível mais baixo. Porém, torna o conjunto de instruções mais próximo das instruções de hardware facilitando a melhoria do desempenho do programa.

1.1.3 Requisitos para o desenvolvimento

Por se tratar de um sistema embutido, para o desenvolvimento do projeto, são necessárias algumas ferramentas básicas como Editores integrados (*IDE Integrated Development Environment*), compiladores (Pacotes para desenvolvimento), simuladores, software de comunicação com o cartão além dos aparatos físicos como cartões de desenvolvimento e dispositivos aceitadores de cartões. Durante o desenvolvimento deste projeto foram usadas as seguintes ferramentas:

- IDEs: NetBeans (NETBEANS, 2012), Eclipse (ECLIPSE, 2012);
- Compiladores: *Java Development Kit* (JDK 1.3.1) (JDK, 2012) e *Java Card Development Kit* (JCDK 2.1.2) (JCDK, 2012);
- Simuladores: NetBeans, JCWDE (Java Card *Workstation Development Environment*) (JCWDE, 2012);
- Software de comunicação: GPShell (Shell para a Global Plataform) (GPSHELL, 2012);
- Cartões de desenvolvimento: JCOP21;
- Dispositivos: Leitor e gravador de cartões de um computador (basta que possua compatibilidade com o padrão PC/SC).

Além das ferramentas comerciais já existentes, viu-se a necessidade de desenvolver uma ferramenta para auxiliar no desenvolvimento e no teste das biometrias com processamento em *Smart Card*, objetivo principal do presente projeto. A ferramenta foi desenvolvida baseada na generalização que é possível se fazer entre diferentes biometrias mudando apenas poucas informações que varias entre elas. A Seção 1.3 traz maiores detalhes acerca desta ferramenta.

1.2 Biometria

Tecnologias biométricas são definidas como métodos automatizados de identificação e/ou verificação de características únicas de um ser vivo podendo essas serem características físicas ou comportamentais. Biometrias são altamente seguras e convenientes para a

identificação ou verificação de identidade de um indivíduo, pois não podem ser roubadas ou esquecidas além da alta dificuldade para forjá-las (COUNCIL, 2012). São diversas as biometrias existentes em estudos. Impressão digital, íris, geometria da mão ou facial, veias da mão e voz são exemplos de características físicas enquanto assinatura e cadência de digitação em teclados são características comportamentais.

A escolha de uma determinada biometria deve levar em conta muitos fatores como a facilidade da coleta, o desempenho da tecnologia, o custo, perfil e a cultura do usuários. Fatores estes que também podem afetar a aceitação da biometria. A Tabela 1[1] mostra um comparativo entre diferentes biometrias em relação às suas características.

Tabela 1: Comparação entre tecnologias biométricas

Identificador biométrico	Universalidade	Distinção	Permanência	Coletabilidade	Desempenho	Aceitabilidade	Chance de fraude
DNA	A	A	A	B	A	B	B
Orelha	M	M	A	M	M	A	M
Face	A	B	M	A	B	A	A
Termograma facial	A	A	B	A	M	A	B
Impressão digital	M	A	A	M	A	M	M
Forma de andar	M	B	B	A	B	A	M
Geometria da mão	M	M	M	A	M	M	M
Veias da mão	M	M	M	M	M	M	B
Íris	A	A	A	M	A	B	B
Dinâmica da digitação	B	B	B	M	B	M	M
Cheiro	A	A	A	M	B	M	M
Impressão da palma da mão	M	A	A	M	A	M	M
Retina	A	A	M	B	A	B	B
Assinatura	B	B	B	A	B	A	A
Voz	M	A	A	M	B	A	A

Uma das características chaves para a seleção do uso de uma biometria específica é a distinção que irá afetar diretamente na acurácia do sistema. Levando em consideração todos os aspectos citados, foram escolhidas as biometrias da Impressão digital, Íris e Impressão da palma da mão. Elas são introduzidas nas seções 1.2.1, 1.2.3 e 1.2.2, respectivamente.

[1]Tabela retirada do artigo (JAIN; ROSS; PRABHAKAR, 2004), onde B, M e A representam baixo, médio e alto, respectivamente. Os resultados foram baseados na percepção dos autores.

1.2.1 Impressão digital

A impressão digital tem sido utilizada na identificação de indivíduos há mais de um século principalmente na área de criminalística e forense. Existem imensos bancos de dados ao redor do mundo como o da FBI (*Federal Bureau of Investigation*), que possui o maior volume de dados do mundo, contendo mais de 200 milhões de impressões digitais. Por conta desse uso, trata-se da biometria mais antiga.

Um outro uso mais atual é o biométrico para a identificação e autenticação da identidade. Uso esse de extremo interesse para este projeto. Atualmente, é a biometria mais usada em diversas áreas. Simplicidade, segurança e facilidade de extração permitem colocá-la a frente de diversos outros tipos de autenticação. Vários métodos foram propostos para a comparação de impressões digitais e o mais comumente utilizado deles é o de comparação de minucias (*minutiae*) (YAGER; AMIN, 2004).

Figura 2: Impressão digital e minúcias

Minúcias (Figura 2) são as características de uma impressão digital quando são analisadas as linhas da impressão. Esse conjunto tende a ser único em cada indivíduo. A comparação de minucias não é uma tarefa simples, principalmente quando é necessário ser executada em um sistema embutido com baixo poder de processamento e pouca memória disponível em relação aos computadores pessoais (PC).

1.2.2 Impressão da palma da mão

A impressão da palma da mão vem sendo usada como um identificador humano há mais de 100 anos e ainda é considerada como uma das formas mais confiáveis de se distinguir uma pessoa devido a sua estabilidade e unicidade (SHU; ZHANG, 1998). Apenas recentemente começou-se a estudá-la para o uso biométrico. Como será visto na Seção 2.2 já existem inúmeros métodos baseados em diversas abordagens.

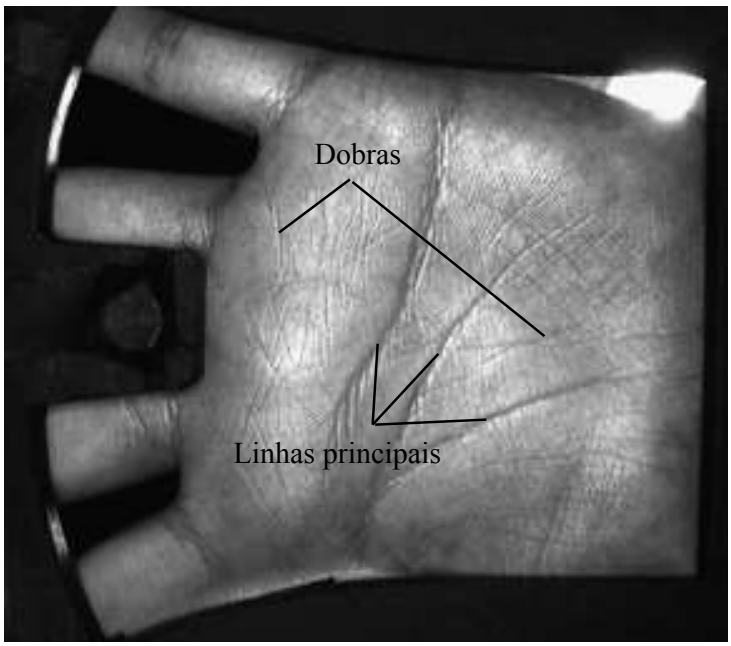


Figura 3: Impressão da palma da mão

Como ilustrado na Figura 3, as principais características utilizadas na biometria da impressão da palma da mão são as linhas principais e as dobras secundárias. Assim como a impressão digital, a impressão da palma da mão também possui linhas e minúcias que podem ser utilizadas para distinguir dois indivíduos. Devido a extensão da impressão da palma da mão, o uso de linhas e minúcias só podem ser utilizados com imagens de sensores de alta resolução. Ainda assim, alguns métodos vão além e utilizam os poros para fazer a distinção.

Um dispositivo usado para a aquisição das imagens da impressão da palma da mão pode ser visto na Figura 4. O dispositivo é capaz de fazer a aquisição da imagem 2D e mapa 3D da mão. A impressão da palma da mão ainda não possui grande penetração no mercado mas, devido aos resultados obtidos por pesquisadores, pode se tornar uma biometria altamente utilizada em breve.

(a) *Dispositivo para aquisição* (b) *Dispositivo em uso*

Figura 4: Dispositivo usado para aquisição da impressão da palma da mão.

1.2.3 Íris

A biometria da íris é uma das mais confiáveis por se tratar de um órgão interno e praticamente invariante durante a vida inteira. É um órgão plano e seu diâmetro é alterado apenas com a contração e dilatação da pupila. Trata-se de uma biometria recente pois seu uso começou a ser difundir em 1993 pelo trabalho do Professor John Daugman (DAUGMAN, 1993). A maioria dos sistemas biométricos atuais são baseados no trabalho de Daugman. A biometria já é utilizada em diversos sistemas de segurança pelo mundo como Emirados Árabes Unidos, Amsterdam Airport Schiphol, Holanda, Canadian Air Transport Security Authority, entre outros.

A íris possui uma textura que é determinada aleatoriamente na fase embrionária, assim como as impressões digitais, e provar que ela é única é praticamente impossível. Entretanto, são tantos os fatores envolvidos em sua formação que a chance de uma verificação falsa ser validada é mínima. A Figura 5 mostra as partes do olho que precisam ser consideradas na comparação da íris.

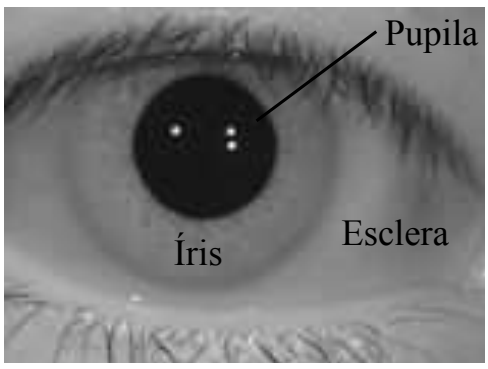


Figura 5: Estrutura do olho

Outro benefício do uso da biometria da íris é a distância de captação, não sendo necessário o toque no equipamento. A maioria dos aparelhos opera entre 10 cm à poucos metros. O benefício também traz uma dificuldade para o uso da biometria. Essa distância, que por um lado é confortável, faz com que seja necessário o tratamento da imagem para a obtenção da íris, o que pode ser uma fonte de erro. Outra fonte de erro vem do número de obstáculos que pode impedir a captação correta da íris como as pálpebras e os cílios.

1.3 Biometria em Cartões Inteligentes

Esta Seção define os aspectos que utilizando cartões inteligentes deverão apresentar os sistemas biométricos implementados. Apesar de abordar a importância destes sistemas, o foco do projeto desta dissertação não é a confecção de um sistema biométrico completo mas sim estudar a viabilidade da implementação de comparações biométricas processadas em cartões inteligentes. Os sistemas biométricos são compostos basicamente por 4 componentes:

- Uma máquina ou mecanismo responsável pela representação digital das características biométricas de uma pessoa;
- Ferramenta de extração do padrão que será usado na comparação;
- Ferramenta de verificação entre o padrão armazenado e o padrão de entrada;
- Interface para a comunicação do resultado.

Os sistemas biométricos operam em dois estágios: o armazenamento do padrão que servirá como base para as comparações e a verificação entre os dados armazenados e os dados de entrada que estão sendo comparados.

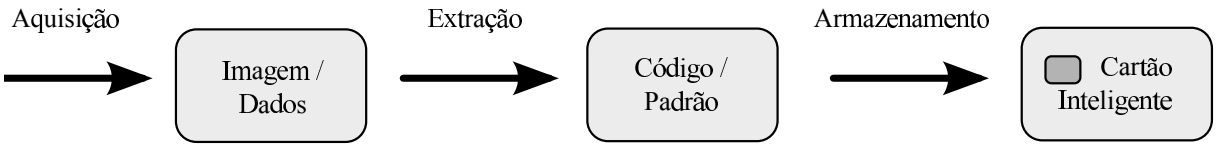


Figura 6: Armazenamento de uma biometria em um Cartão Inteligente

A Figura 6 ilustra o processo de armazenamento (*Enrollment*). A amostra do indivíduo, usuário do cartão, é capturada. Para cada biometria um método específico

será utilizado (scanner para impressões digitais, microfone para reconhecimento de voz, câmera para reconhecimento de face, câmera para reconhecimento de íris, etc.). Os dados coletados são então processados para a extração das características únicas do usuário. O padrão biométrico extraído que será usado nas futuras comparações é armazenado no cartão.

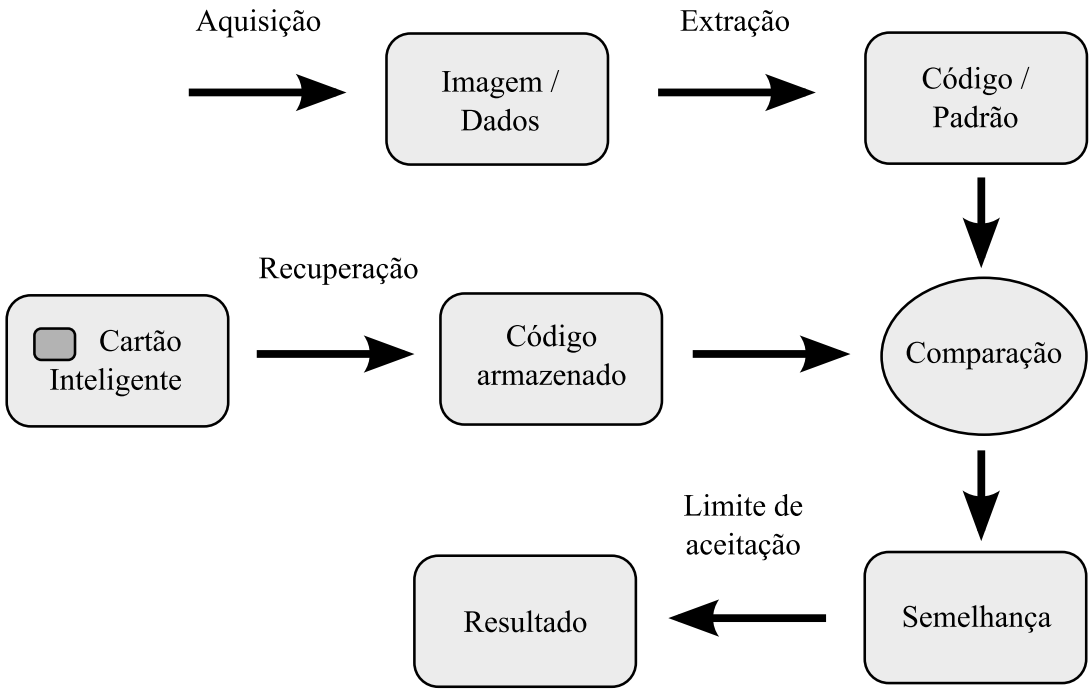


Figura 7: Verificação biométrica

A Figura 7 ilustra o processo de verificação biométrica (*Matching*). A amostra biométrica do requerente é capturada de forma semelhante à captura realizada durante o estágio de armazenamento. Os padrões únicos dessa amostra são extraídos e enviados ao comparador. O padrão armazenado é recuperado do cartão e enviado ao comparador que então processa a verificação que resulta em uma nota, definindo se as duas amostras biométricas são do mesmo indivíduo ou não. Sistemas biométricos podem objetivar a identificação e/ou verificação. A identificação é a procura por uma pessoa a partir de uma dada amostra biométrica. Esta exige grandes bancos de dados e muito poder de processamento. Técnicas de indexação para melhorar o sistema de busca são necessárias. A verificação é a validação entre duas amostras, resultando na identificação se a amostra

é da mesma pessoa ou não. As comparações biométricas usando cartões podem ocorrer de duas formas:

- *Template on Card* (ToC), onde o padrão do usuário é guardado na memória do cartão e a comparação é feita externamente em outra máquina. Isso necessita apenas cartões com memória, que são mais baratos.

- *Match on Card* (MoC), onde o padrão do usuário é guardado na memória do cartão e a comparação é processada também no cartão. São necessários cartões dotados de pelo menos um processador. O baixo poder de processamento e o tamanho reduzido da memória são os maiores obstáculos. Neste projeto será feita a comparação entre três algoritmos de verificação biomética (impressão digital, íris e impressão da palma da mão) que serão processados no cartão para posterior análise e comparação.

Em um sistema biométrico, quando a informação armazenada é comparada à informação capturada, uma nota de similaridade é atribuída e usada para confirmar a identidade de um indivíduo. Quando essa nota é comparada com um dado limite, dois tipos de taxa de erro podem ser observados:

- Taxa de aceitação incorreta (FAR - *False Acceptance Rate*), que indica a taxa de entradas falsas ou impostoras incorretamente aceitas.

- Taxa de rejeição incorreta (FRR - *False Rejection Rate*), que indica a taxa de entradas do indivíduo correto incorretamente rejeitadas.

Essas duas taxas são de extrema importância na escolha do limite da nota que deverá definir a escolha das comparações que serão declaradas como falsas ou verdadeiras. Quando se trata de sistemas embutidos, um fator de extrema importância também é a escolha do algoritmo pois é necessário averiguar a complexidade, o uso de memória e tempo de execução do mesmo. Para a escolha das biometrias a serem implementadas no projeto desta dissertação também foram considerados a forma de extração e disponibilidade de uma ferramenta para esse fim.

Tendo em vista os aspectos comuns à todas as biometrias, foi desenvolvida uma ferramenta unificada de testes que foi usada para testar todas as biometrias desenvolvidas. A Seção 1.3.1 irá introduzir esta ferramenta.

1.3.1 Software de testes

A ferramenta foi desenvolvida no intuito de auxiliar tanto no desenvolvimento das biometrias dentro dos cartões quanto nos testes das mesmas. A Figura 8 traz a interface do software desenvolvido especificamente para o projeto desta dissertação.

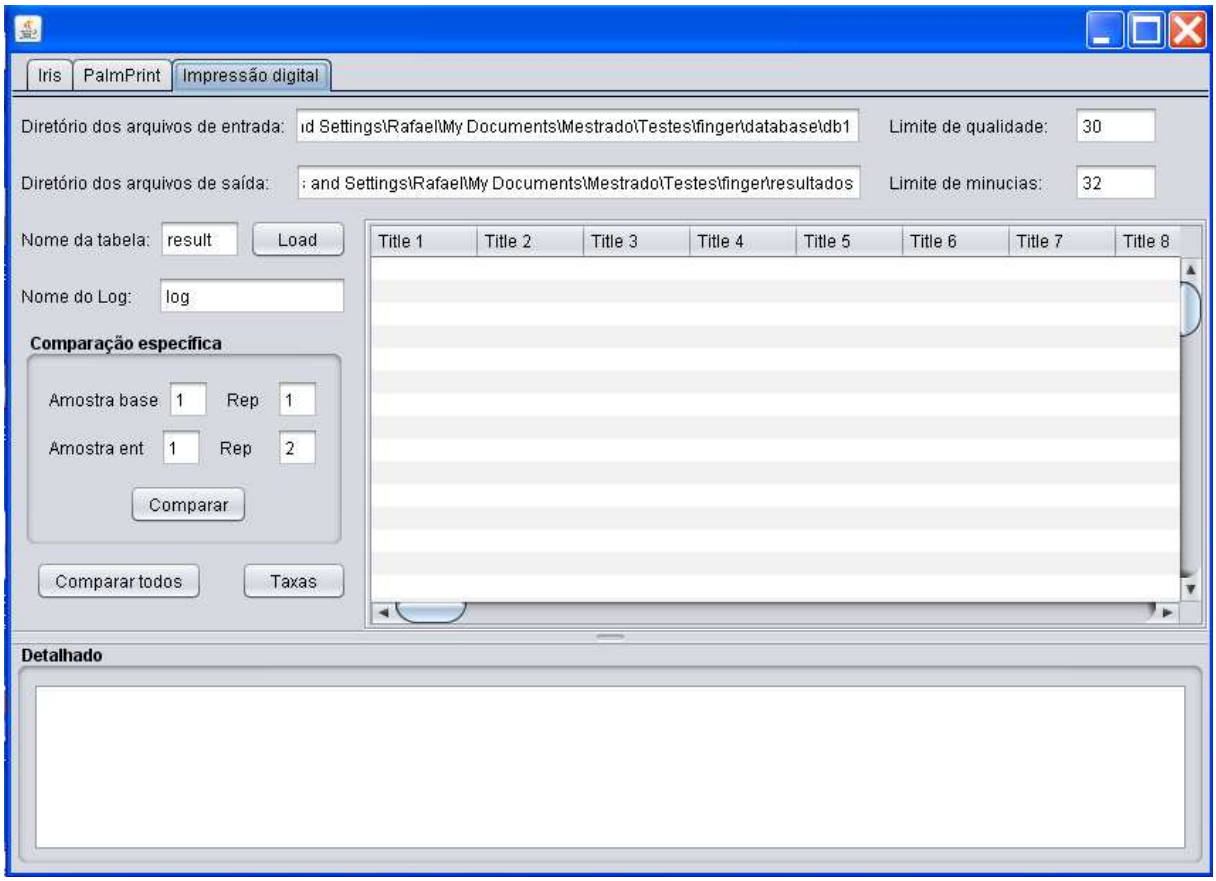


Figura 8: Interface da ferramenta

O software desempenha diversas funções e algumas são específicas de apenas uma das biometrias. Segue uma lista com a descrição de todos os campos utilizados pelo software.

- Abas: O software possui 3 abas sendo uma para cada biometria desenvolvida. pode ser expandido para mais abas pois possui um desenvolvimento generalizado facilitando a inserção de novas biometrias.
- Diretório de arquivos de entrada: Trata-se do diretório que contém os arquivos que guardam os códigos das biometrias a serem usadas nos testes. A forma como o código de cada biometria é gerado será vista nos capítulos específicos.

- Diretório de arquivos de saída: Indica o diretório onde os arquivos contendo os resultados dos teste deverão ser gerados.

- Limite de qualidade e Limite de minúcias: Esses são campos específicos da biometria da impressão digital e indicam a qualidade mínima e a quantidade máxima de minúcias contidas no arquivo de entrada que serão consideradas.

- Nome da tabela: Nome do arquivo que receberá o arquivos contendo o resultado dos testes e será salvo no diretório dos arquivos de saída.

- Nome do Log: Nome do arquivo de depuração.

- Campos do grupo Comparação específica: Muito útil para o desenvolvimento da biometria pois compara dentro do cartão as duas amostras indicadas sendo a amostra base usada como biometria do indivíduo dono do cartão.

- Botão Comparar todos: Serve para comparar automaticamente todas as amostras contidas no diretório de entrada, ou seja, cada amostra é comparada com todas as demais.

- Botão Taxas: É utilizado após feito o teste que compara todos. Gera arquivos com dados calculados usando o resultado das comparações. São gerados automaticamente as taxas de falso positivo e falso negativo que posteriormente foram usados para a comparação entre as diferentes biometrias.

- Tabela: Contém o resultado das comparações entre as amostras.

- Detalhado: Contém informações adicionais como as mensagens trocadas com o cartão e o tempo da execução do comando indicado.

O Software mostrou-se extremamente valioso para a realização automática dos testes entre as diversas amostras de cada uma das biometrias. Em alguns casos foram usadas 200 amostras, ou seja, 40.000 comparações possíveis. Sem a existência deste software, o processo de teste seria extremamente penoso ou até inviável e comprometeria a qualidade dos resultados apresentados.

1.4 Considerações Finais

Neste capítulo foram vistas as principais características dos *Smart Cards* e biometrias que serão usados em conjunto na tentativa de obter uma ferramenta de segurança com maior potencial. A plataforma Java Card foi escolhida para a implementação do projeto devido ao grande número de ferramentas e especificações disponíveis além de conter todas as características desejáveis.

As biometrias a serem implementadas foram escolhidas devido a comparação mostrada na Seção 1.2. Todo o trabalho desenvolvido para as biometrias da Impressão digital, Impressão da palma da mão e Íris será detalhado nos Capítulos 3, 4 e 5, respectivamente.

Capítulo 2

TRABALHOS RELACIONADOS

ESTE capítulo relaciona alguns trabalhos correlatos ao tema do projeto desta dissertação. Serão brevemente descritos os principais trabalhos relacionados às biometrias da impressão digital, da impressão da palma da mão e da íris nas Seções 2.1, 2.2 e 2.3 respectivamente. Esses trabalhos serviram de base para a escolha do algoritmo mais adequado à implementação em um cartão inteligente.

2.1 Impressão Digital

A maioria das técnicas de reconhecimento e algoritmos de comparação de impressões digitais se baseia em características presentes nas linhas da impressão. Essas características receberam o nome de minúcias (*minutiae*). Uma das razões desse uso se dá pelo longo histórico da impressão digital em diversas áreas nos quais especialistas humanos também são responsáveis pela comparação (YAGER; AMIN, 2004). A Seção 2.1.1 relaciona alguns trabalhos com métodos baseados em minúcias enquanto a Seção 2.1.2 listará outros que não são baseados em minúcias.

2.1.1 Métodos utilizando minúcias

Minúcias são pontos característicos presentes na impressão digital. Cada minúcia representa um final ou uma bifurcação de linha da impressão. São identificadas por sua localização e o ângulo da linha onde está presente. Para a correta verificação das minúcias é essencial que o maior número possível de minúcias seja extraído com alta confiabilidade. Na Seção 2.1.1.1, são apresentados alguns métodos de extração da minúcia enquanto na Seção 2.1.1.2 são explicados alguns métodos de comparação das minúcias.

2.1.1.1 Extração

Após a aquisição da imagem da impressão digital, é necessário filtrá-la antes de extrair as minúcias. Diversos filtros podem ser usados para realizar o tratamento e afinamento das linhas da impressão digital. Isto que consiste basicamente em fazer com que a largura das linhas tenha apenas um pixel em uma imagem em preto e branco. A escolha do filtro depende principalmente da qualidade e do método de captura da imagem da impressão digital. Antes do afinamento, é necessário estimar um campo de orientação das linhas da impressão. Em (JAIN; HONG; BOLLE, 1997) e (RAO, 1990) é apresentado um método para o cálculo desse campo de orientação em que a imagem é dividida em blocos iguais de tamanho pré-definido e cada um deles tem sua orientação estimada.

Em (JAIN et al., 1997), após encontrar o campo de orientação da imagem da impressão digital, é realizada a convolução da imagem usando duas máscaras para acentuar os níveis de cinza das linhas da imagem. Com isso, é possível determinar a localização das linhas observando os máximos locais. Para extrair as linhas é aplicado um limite que define o que é *linha* e o que é *fenda*. Os *pixels* que apresentarem um nível de cinza acima do limite serão trocados por 1 (branco) enquanto aqueles com nível de cinza abaixo do limite serão substituídos por 0 (preto).

Os autores de (CHANG; FAN, 2001) propuseram um método para a detecção de linhas usando um algoritmo mais complexo baseado na decomposição do histograma de níveis de cinza. Apesar da complexidade, o método proposto apresentou melhores resultados do que outros métodos quando a imagem é de baixa qualidade. Após separar o que representa linha do resto, é necessário fazer o afinamento. Apesar de parecer simples, os resultados da etapa de afinamento podem influenciar a próxima etapa. Algumas falhas na imagem podem ser erroneamente interpretadas e irão produzir minúcias de baixa qualidade na próxima etapa. Essa etapa e seus problemas são abordados com maiores detalhes em (FITZ; GREEN, 1996) e (RAO, 1976).

Após o afinamento, ou seja, linhas com largura de um pixel, a extração das minúcias pode ser abordada de forma relativamente simples, bastando estudar a vizinhança de cada pixel preto. Se a vizinhança apresentar apenas um *pixel* em preto, esse *pixel* será considerado uma minúcia de final de linha, se apresentar dois *pixels* em preto, será um ponto de uma linha e se apresentarem três *pixels* em preto, será uma bifurcação. Entretanto, essa abordagem simples faz com que linhas partidas, linhas dobradas, entre outros

casos, sejam incorretamente interpretadas como minúcia e dessa forma será detectado um número maior de minúcias do que realmente existe (FARINA; VAJNA; LEONE, 1999). Dada a importância da extração, é necessário aplicar filtragens adicionais para impedir a ocorrência desse problema.

Em (CHEN; KUO, 1991), os autores usam diversos métodos heurísticos para eliminar minúcias falsas. Aplicam limites para remover pontas de linhas muito curtas. Para corrigir quebras de linhas, espaços pequenos entre duas linhas de mesma direção são eliminados. Nos casos de múltipla detecção de minúcias em uma pequena área, a chance de sucesso é menor por se tratar de uma provável área de ruído. Em (RATHA; CHEN; JAIN, 1995) são usados operadores morfológicos para detectar e remover pontas. Vários outros métodos são utilizados a fim de eliminar falsas minúcias (XIAO; RAAFAT, 1991), (HUNG, 1993) e (FARINA; VAJNA; LEONE, 1999). Mesmo com o uso de métodos sofisticados não é incomum a presença de falsas minúcias. Em (MAIO; MALTONI, 1997) e (MAIO; MALTONI, 1998) são usadas redes neurais artificiais para filtrar falsas minúcias.

2.1.1.2 Comparação

As minúcias podem ser classificadas entre final de linha ou bifurcação mas normalmente essa diferenciação não é levada em conta por ser comum que extratores invertam essa classificação. Essa diferenciação pode melhorar o desempenho do algoritmo como visto em (PRABHAKAR; JAIN; PANKANTI, 2003) mas torna o processo ainda mais dependente da qualidade da extração tendo melhores resultados com imagens de resolução alta.

O resultado da comparação entre as minúcias de diferentes impressões digitais é normalmente dado pela distância entre elas. Essa forma simples de comparação é introduzida em (JAIN; HONG; BOLLE, 1997). Mas antes que as minúcias possam ser comparadas, é necessário alinhá-las de forma que seja possível a comparação. Essa fase de alinhamento é conhecida como *registro*. Em (RATHA et al., 1996), a melhor rotação e translação são estimadas usando uma transformada de Hough generalizada. Todos os pares de uma minúcia de uma das impressões com uma minúcia da outra impressão são comparados de forma a guardar a rotação e translação necessárias para que elas representem a mesma minúcia. Todas essas representações são guardadas em um acumulador e o par rotação e translação mais repetido é considerado como o melhor alinhamento. Em (CHOUTA et al., 2012) uma abordagem semelhante é empregada mas também são consideradas as vizinhanças de cada rotação e translação. Dessa forma, as deformações da elasticidade da pele são levadas em

conta. Além disso, o autor propõe o uso de sub-espaços para diminuir a necessidade de memória para guardar todos os valores de translação e rotação.

Uma representação da impressão digital usando grafos foi proposta em (ISENOR; ZAKY, 1986), onde os nós representam as linhas da impressão digital e são conectadas a outros nós quando as linhas são vizinhas diretas. O método não faz a extração explícita das minúcias mas elas acabam sendo representadas como parte do grafo quando dois nós são ligados por dois caminhos diferentes, sendo um diretamente e outro por meio de um nó. Esse método é invariante quanto a rotação e translação uma vez que só leva em consideração a relação entre as linhas e as linhas vizinhas. Apesar de se mostrar interessante em alguns aspectos, o método não é robusto contra falhas nas extração dos grafos e a própria comparação de grafos é uma operação computacionalmente complexa.

Em (JAIN et al., 1997) e (JAIN; HONG; BOLLE, 1997) é usada a forma e localização da linha de onde a minúcia foi extraída para auxiliar no processo de alinhamento. Apesar dos bons resultados, é exigido um alto poder computacional e o método se torna vulnerável a pequenas deformações das linhas uma vez que pequenas distorções podem levar a alinhar incorretamente as minúcias.

Um dos problemas que pode afetar o desempenho da comparação de minúcias é a elasticidade natural da pele. Foi proposto em (CAPPELLI; MAIO; MALTONI, 2001) um modelo para a distorção elástica da impressão digital e incorporado ao algoritmo de comparação das impressões.

2.1.2 Métodos que não utilizam minúcias

Apesar da grande importância das minúcias, diversos métodos foram propostos para uso na biometria da impressão digital. A estrutura cíclica de regiões da impressão (HATANO et al., 2002), assinatura da forma das linhas da impressão (CEGUERRA; KOPRINSKA, 2002), histogramas dos micro-padrões direcionais (WANG; LEE, 1999) também foram usados como características. Alguns autores também usaram *wavelets* (LEE; CHUNG, 1997), (LEE; NAM, 1999) e (TICO et al., 2001) e filtro de Gabor (JAIN et al., 2000), (LEE; WANG, 2001) e (ROSS; REISMAN; JAIN, 2002) para extração de características.

Coeficientes de *wavelets* são conhecidos pela sensibilidade à rotação e translação. Em (TICO et al., 2001), é verificado que esse método possui um potencial limitado para a comparação de impressões digitais por conta desta sensibilidade. Em (JAIN et al., 2000),

foram obtidos resultados mais sólidos usando filtro de Gabor para extrair as linhas da impressão digital.

A tendência é que o uso de minúcias continue sendo o principal método para AFIS (*Automated Fingerprint Identification System*) de alto desempenho (YAGER; AMIN, 2004). Entretanto, os métodos que não usam minúcias podem ser usados de forma suplementar. Em (PRABHAKAR; JAIN, 2001) foram combinados 3 métodos baseados em minúcias e um extrator de texturas usando filtro de Gabor e obteve-se significativa melhora nos índices de erro.

2.2 Impressão da Palma da Mão

A biometria da impressão da palma da mão (*palmprint*) começou a ser estudada recentemente mas tem se mostrado uma tecnologia biométrica promissora (ZHANG; ZUO; YUE, 2012). A palma da mão é uma superfície muito rica em detalhes, desde os mais destacados, como as linhas principais, até os menores detalhes, como as minúcias estudadas na impressão digital ou até mesmo os poros da palma da mão.

Para o caso de comparações em cartões, os métodos que utilizam os menores detalhes (minúcias e poros) não são interessantes, uma vez que haverá a necessidade de sensores de alta resolução e o número de detalhes extraídos será muito grande necessitando, posteriormente, de mais processamento na fase comparação. Logo, são considerados nesta seção métodos que utilizam imagens de baixa resolução (menos do que 100dpi).

Atualmente existem 3 principais tipos de abordagem de reconhecimento de impressão da palma da mão. São elas holística, de características específicas e híbrida.

2.2.1 Abordagem holística

Nesta abordagem, a imagem da impressão da palma da mão é usada como base em um extrator ou classificador holístico. Para seu uso, existem dois principais problemas: a representação da imagem e o projeto do classificador. A Seção 2.2.1.1 introduz as representações mais usadas enquanto a Seção 2.2.1.2 trata dos classificadores.

2.2.1.1 Representação

Imagens da impressão da palma da mão podem ser representadas tanto no domínio do espaço como em outros domínios transformados. As características holísticas podem ser

extraídas usando essas representações de diversas maneiras.

Concatenando colunas de uma imagem de impressão da palma da mão em um vetor de várias dimensões, são feitas analises de uma variedade de subespaços lineares e não lineares para a extração das características (LU; ZHANG; WANG, 2003), (WU; ZHANG; WANG, 2003) e (YANG et al., 2007). Recentemente, foram desenvolvidos analisadores de tensão tratando a imagem da impressão como tensores de segunda ordem (HU; FENG; ZHOU, 2007) e (ZUO; ZHANG; WANG, 2006).

Foram investigadas diferentes técnicas de transformações comuns em processamento digital de imagens para a representação de uma imagem de impressão da palma da mão. A transformada de Fourier, técnica de transformação clássica para imagem, foi aplicada com sucesso para extração de características e projeto de classificador (JING; TANG; ZHANG, 2005) e (LI; ZHANG; XU, 2002).

2.2.1.2 Classificação

Em (HAN et al., 2003), uma rede neural de retro-propagação (*backpropagation*) foi inicialmente aplicada para em autenticação da impressão da palma da mão entretanto o reconhecimento da palma da mão é um problema típico de multi-classes o que o torna muito difícil para redes de retro-propagação.

Em (LI; WANG; ZHANG, 2005), a rede neural modular é usada para decompor a tarefa de reconhecimento da impressão da palma da mão em uma série de menores e mais simples subproblemas de duas classes.

2.2.2 Abordagem das características locais

Existe uma série de características da palma da mão que podem ser utilizadas para seu reconhecimento. A Tabela 2 lista as principais em termos de resolução necessária, coletabilidade, permanência e distinção.

Tabela 2: Características específicas no reconhecimento da impressão da palma da mão

Característica	Resolução	Coletabilidade	Permanência	Distinção
Linhas principais	baixa	alta	alta	baixa
Dobras	média	alta	média	alta
3D	média	baixa	média	média
Minucias	alta	média	alta	alta
Nível 3	Muito alta	baixa	média	alta

As linhas principais não são confiáveis para realizar uma comparação direta mas podem ser usadas para obter pré-alinhamento das imagens antes da comparação mais detalhada como usado em (LI et al., 2012). Normalmente, as linhas principais são usadas em conjunto com outras características para aumentar a confiabilidade do resultado da comparação.

As dobras da palma da mão podem ficar semelhantes por meses ou anos mas não são permanentes como as minucias. Por esse motivo, não são úteis em áreas mais críticas como criminalística e forense mas podem ser usados em sistemas de reconhecimento em tempo real, estabelecendo alto desempenho (SUN et al., 2005) e (ZHANG et al., 2003).

Em (ZHANG et al., 2009), a estrutura 3D da palma da mão é utilizada para aumentar a confiabilidade do reconhecimento e combater ataques usando falsas impressões. Apesar de aumentar também a dificuldade da aquisição dos dados, quando aliado à textura 2D, o processo se torna altamente confiável e robusto contra fraudes.

O uso de minúcias tem mostrado recentemente grande potencial na área forense e criminalística (JAIN; FENG, 2009). Para seu correto funcionamento são necessárias imagens com resolução mínima de 500dpi. Uma grande vantagem da utilização de minúcias é a possibilidade de realizar o reconhecimento com um alto grau de confiabilidade utilizando apenas uma parte da impressão da palma da mão.

Características de nível 3 foi o nome dado ao conjunto de características que engloba todas as anteriores e os mínimos detalhes da palma da mão como linhas da impressão, poros e cicatrizes (JAIN; CHEN; DEMIRKUS, 2007). O uso de características de nível 3 são ainda mais importantes para a identificação usando apenas parte da impressão da palma da mão mas, para isso, devem ser usadas imagens com alta resolução (maior do que 1000dpi). Estima-se que 20 à 40 poros são suficientes para identificar um indivíduo (ASHBAUGH, 1999).

Como mencionado anteriormente, foram considerados apenas os métodos que utilizam imagens de baixa resolução. Logo, os algoritmos estudados utilizam as linhas principais e dobras da palma da mão. Existem três principais mecanismos de extração e comparação. Podem se basear em *linhas*, em *códigos* ou em *descritores da textura* da palma da mão.

2.2.2.1 Linhas

Em (WU; ZHANG; WANG, 2006b), é usada a derivada de segunda ordem da Gaussiana para representar a magnitude da linha e a de primeira ordem para detectar a localização da linha. Todas as linhas direcionais são combinadas para formar o resultado final.

O problema desse tipo de abordagem é o fato de ser inevitável passar por um processo de alinhamento por rotação e translação durante a comparação. Processo esse que precisa de um processamento elevado. Para contornar esse problemas e aumentar o desempenho, o autor dilatou as linhas extraídas antes da comparação.

2.2.2.2 Códigos

Os métodos baseados em códigos transformam a imagem resultante da passagem de alguns filtros em códigos binários. Usando códigos binários, são obtidas algumas vantagens como baixa necessidade de memória e a comparação rápida. Por isso, esses códigos têm sido muito útil na representação e comparação de impressões de palma da mão.

Inspirados no *IrisCode* (DAUGMAN, 1993), foi desenvolvido em (ZHANG et al., 2003), o método *PalmCode*. Inicialmente, é feita a convolução da imagem de impressão da palma da mão usando um filtro 2D de Gabor e então as imagens real e imaginária resultantes são codificadas de acordo com a sua fase em uma representação binária. Para melhorar o desempenho do método, é possível extrair vários *PalmCodes* usando filtros de Gabor em diversas orientações. Dessa forma foi desenvolvido o método *FusionCode* (KONG; ZHANG; KAMEL, 2006), que faz com que ocorra a diminuição das taxas de erro.

Recentes avanços no estudo de métodos usando códigos indicam que uma das características mais promissoras para o reconhecimento da impressão da palma da mão é a orientação das linhas (KONG; ZHANG, 2004) e (WU; ZHANG; WANG, 2006a). Tais métodos foram capazes de alcançar taxas de erro praticamente nulas.

2.2.2.3 Descritores de textura

Um descritor de texturas da impressão da palma da mão típico divide a imagem da palma da mão em pequenos blocos, então calcula a média, variância, energia ou histograma associados à cada um desses blocos, como características locais (HAN; TAN; SUN, 2007), (WANG et al., 2006), (KUMAR; ZHANG, 2006) e (WU; WANG; ZHANG, 2002).

Em (KUMAR; ZHANG, 2006), a imagem da impressão da palma da mão é divida em blocos sobrepostos, são calculados os coeficientes DCT (*Discrete Cosine Transform*) associados à cada bloco e seus desvios padrão são usados para formar um vetor de características.

Outros descritores de textura como energia do elemento direcional e histograma da direção local também foram adotadas no reconhecimento da impressão da palma da mão (HAN; TAN; SUN, 2007) e (WU; WANG; ZHANG, 2002).

2.2.3 Abordagem híbrida

Vem sendo discutido que o sistema de visão humano faz uso de ambas as características locais e de percepção holística para reconhecer e identificar objetos de interesse e é esperado que sistemas com abordagens híbrida desse tipo sejam promissores para o reconhecimento de impressões da palma da mão (ZHANG; ZUO; YUE, 2012). Sistemas híbridos tem duas principais aplicações: reconhecimento de alta acurácia (KUMAR; ZHANG, 2005) e rápida identificação de impressões (LI; YOU; ZHANG, 2005) e (YOU et al., 2004).

Usando ambas as abordagens para obter múltiplas representações da impressão da palma da mão pode-se usar várias estratégias para a fusão de tipos de características, decisões e notas para melhorar o desempenho das comparações (KITTLER et al., 1998). Nesse caminho, um grande número de abordagens usando múltiplas impressões foi sugerido em (KUMAR; ZHANG, 2005) e (POON; WONG; SHEN, 2004). Em (KUMAR; ZHANG, 2005) são extraídas três representações: Gabor, linha e características de subespaços. Então, foi proposto um produto entre os resultados para ter uma combinação das suas notas de comparação.

Em (YOU et al., 2004) são usados vários níveis para a identificação da impressão da palma da mão: geometria da mão (nível 1), energia da textura global (nível 2), características *fuzzy* das linhas (nível 3) e energia da textura local (nível 4) para serem usados como classificadores hierárquicos. É proposto então um esquema de busca guiada para obter uma comparação eficiente. Esse método pode ser usado tanto para o reconhecimento quanto para a identificação de um usuário. Partindo do nível mais baixo, as impressões são comparadas usando um limite pré-estabelecido, até a última comparação do nível mais alto para garantir a autenticidade do usuário.

2.3 Íris

A mais importante das primeiras publicações a respeito de métodos de reconhecimento pessoal pela biometria da íris pertence ao professor Daugman (DAUGMAN, 1993). A técnica proposta em (DAUGMAN, 1993) descreve com certos detalhes os processos de segmentação, extração e comparação. O trabalho de Daugman se tornou a maior referência nesse segmento e serviu de base para praticamente todos os modelos biométricos de íris existentes (BOWYER; HOLLINGSWORTH; FLYNN, 2008).

No trabalho de Daugman é detalhado todo o processo de segmentação da íris. É usado um operador integral-diferencial para encontrar a localização da íris assim como as regiões cobertas pelas pálpebras ou por reflexos da luz. Para normalizar o resultado levando em conta diferentes distâncias ou resoluções, são adotadas duas coordenadas: ângulo variando de 0^{o} à 360^{o} e uma coordenada radial que varia de 0 à 1 independente do tamanho da imagem ou da dilatação e contração da pupila, sendo que as deformações ocorridas em decorrência dos dois movimentos foram consideradas lineares. A imagem então é transformada em um retângulo assumindo a coordenada radial como eixo vertical e a coordenada angular como eixo horizontal.

Comparar as imagens de forma bruta levariam a muitos erros devido à influência da luminosidade. Para evitar isso é usada a convolução da imagem, realizada via um filtro Gabor de duas dimensões para extrair as informações da textura. O resultado desta convolução é uma matriz de números complexos que são então codificadas usando apenas suas fases. Resultando em uma matriz de números binários, que representam as fases dos números complexos, com tamanho total de 256 *bytes*.

Para realizar a comparação entre dois códigos é feita também uma máscara que delimita quais são os *pixels* válidos, ou seja, que não estão cobertos por pálpebras ou reflexos. A comparação é feita de forma binária usando a distância de Hamming, uma operação de "ou exclusivo" é aplicada sobre todo o código e o número de 1s, que diferem, é contado e relacionado ao número de *bits* válidos.

Um trabalho conseguiu algum destaque sem usar o trabalho de Daugman como base. Em (WILDES, 1997), diferente de Daugman, é usada a transformada de Hough para detectar os círculos interno e externo da íris. Para comparação, aplicou-se o filtro da Laplaciana da Gaussiana em múltiplas escalas e computou-se a similaridade das imagens. O autor aponta resultados positivos em relação à falsa aceitação, mas indica que o sistema

não é muito flexível em relação ao posicionamento da íris e a luminosidade do ambiente.

Após os primeiros trabalhos na área de biometria da íris, vários outros surgiram para melhorar alguns aspectos, tais como segmentação, extração e comparação dos métodos já existentes. Em (HUANG; DASS; JAIN, 2005), é sugerido uma modificação no processo de Wildes para procurar a íris na imagem em outra escala. É apresentada a ideia única de fazer a comparação usando os dois olhos, sendo o esquerdo para a comparação e o direito para o alinhamento correto.

2.3.1 Segmentação

Em (LIU et al., 2003), é usado o detector de bordas de Canny e a transformada de Hough mas, na tentativa de simplificar o processo, admite-se que a borda da pupila e a íris são concêntricas. Algumas imagens validam essa afirmação porém não pode ser aplicada a todos os casos, pricipalmente, em fotos tiradas em ângulos diferentes. Para melhorar o processo de segmentação pode ser usada uma equalização usando um filtro de passa-alta (SUNG et al., 2004).

Em 2002, Camus e Wildes (CAMUS; WILDES, 2002) apresentam um método que não leva em conta a transformada de Hough, sendo mais semelhante ao método usado por Daugman. As coordenadas e raio de um círculo são modificados de forma a encontrar a melhor solução em um espaço de busca. O algoritmo considera também um limite da relação entre os raios interno e externo da íris. A taxa de acerto foi de 99,5% para pessoas sem usar óculos e 66% para pessoas usando óculos. A grande vantagem do algoritmo foi a diminuição do processamento, i.e. 3,5× mais rápido que o algoritmo proposto por Daugman (DAUGMAN, 2001), que obteve taxas de acerto semelhantes.

2.3.2 Extração

Diferentes filtros foram propostos para a extração de características da íris. Em (SUN; TAN; QIU, 2004), é usado um filtro Gaussiano. É realizada a convolução do campo dos vetores de gradiente da imagem usando o filtro Gaussiano e depois, cada parte do resultado é classificada de acordo com 6 diferentes opções, em contraste com o processo de Daugman em que o resultado da convolução é classificado em relação à fase do número complexo.

Em (CHANGHONG; ZHAOYANG, 2005), é usado o filtro Laplaciano da Gaussiana para efetuar a convolução da imagem e assim extrair as características denominadas

"blobs" que são as áreas mais escuras relativamente às suas vizinhanças. Um código então é construído baseado-se na presença ou ausência de *blobs*. Em (CHOU et al., 2006) também é usado o filtro Laplaciano da Gaussiana em conjunto com o filtro Derivativo da Gaussiana para determinar se um *pixel* é uma borda do tipo "step" ou "ridge". Baseado-se nessas características extraídas, pode ser obtida uma medida de similaridade entre duas íris. A vantagem deste tipo de filtro mais simples do que o filtro de Gabor, é o menor número de parâmetros necessários tornando mais fácil o ajuste. Ambos sugerem o uso de algoritmos genéticos para encontrar os valores mais adequados dos parâmetros.

2.3.3 Comparação

Alguns métodos biométricos utilizam múltiplas amostras como base para melhorar os resultados das comparações. Isso também pode ser feito para métodos biométricos da íris. Em (DU, 2006), são usadas de uma à três imagens como base da operação de comparação, conseguindo uma taxa de acerto de 98,5%, 99,5% e 99,8% no caso de 1, 2 e 3 imagens usadas, respectivamente. Em (MA et al., 2003), sugere-se que a imagem escolhida deve ser aquela com a melhor qualidade e em (MA et al., 2004), é usada uma média das três comparações.

Em (DAUGMAN; DOWNING, 2001), é descrito um experimento para determinar a variação estatística da textura da íris. foram feitas 2,3 milhões de comparações entre diferentes pares de íris. Foi encontrada uma distância de Hamming média, que pode variar de 0 à 1, de 0,499 com um desvio padrão de 0,032. A distribuição se aproximou de uma distribuição binomial com 244 graus de liberdade. Também foi estabelecido que a comparação entre as duas íris de uma mesma pessoa e entre as de pessoas diferentes não tem diferença estatisticamente significativa.

2.4 Considerações Finais

Neste capítulo, foram vistos os principais trabalhos relacionados às biometrias da impressão digital, da impressão da palma da mão e da íris. A maioria dos trabalhos relacionados à biometria da impressão digital utiliza as minúcias para efetuar as comparações. Para a biometria da palma da mão, existem diversos métodos utilizados como os baseads em características locais, descritores de textura e até mesmo os híbridas. Foi mostrado que a

biometria da íris foi inicialmente proposta por Daugman e vários trabalhos foram baseados no dele mas alguns utilizaram ideias diferentes.

Todos os trabalhos foram estudados para que fosse feita a escolha do algoritmo a ser implementado em um cartão inteligente. A escolha do algoritmo utilizado para cada biometria estudada é abordada nos Capítulos 3, 3 e 3 para impressão digital, impressão da palma da mão e íris, respectivamente.

Capítulo 3

IMPRESSÃO DIGITAL

ESTE capítulo traz todos os resultados obtidos para a implementação da biometria da impressão digital em um cartão inteligente. Por se tratar de uma forma de identificação e diferenciação há muito tempo conhecida, são vários os métodos e algoritmos existentes para a comparação entre duas impressões digitais. Como mencionado na Seção 2.1, a maioria dos algoritmos faz a comparação utilizando minúcias e foram os mais estudados para a escolha do algoritmo a ser implementado no cartão inteligente.

O método de extração das minúcias, o algoritmo de comparação e o banco de dados utilizado são abordados nas Seções 3.1, 3.2 e 3.3, respectivamente. Alguns detalhes da implementação em cartões inteligentes utilizando Java Card e os resultados obtidos são discutidos nas Seções 3.4 e 3.5, respectivamente. Vale ressaltar que, neste contexto, somente a comparação foi implementada. A extração das minúcias foi realizada por meio da ferramenta NBIS (*NIST Biometric Image Software*) (NBIS, 2012) conforme detalhado no decorrer deste capítulo.

3.1 Extração

Como mencionado na Seção 1.2.1, a comparação não se dá pela análise das imagens de impressões digitais e sim pelas características pontuais extraídas das mesmas. Essas características são chamadas minúcias (*minutiae*). O processo de extração das minúcias é composto por 4 etapas que antecedem o processo de verificação das impressões digitais. São elas: Aquisição da imagem, representação , pré-processamento e extração das características (YAGER; AMIN, 2004). As duas primeiras etapas serão explicadas na Seção 3.1.1 enquanto as restantes serão vistas, respectivamente, nas Seções 3.1.2 e 3.1.3. As minúcias extraídas ao final do processo devem ser compatíveis com os padrões INCITS 378 e

ISO/IEC 19794.

3.1.1 Aquisição e representação da imagem

Para reconhecer a impressão digital é necessário primeiramente obter a imagem em formato digital. Esse processo pode ser realizado de diversas formas. Uma das formas, ainda muito utilizada, é a cópia usando tinta. É conhecida como aquisição *offline*. Posteriormente essas impressões são digitalizadas. A outra forma ocorre *online*. *Hardware* com sensores capazes de adquirir a imagem são usados. A saída é apresentada como uma imagem digital. Para um sistema de tempo real, obviamente, apenas a aquisição *online* é satisfatória. O tipo de aquisição será importante na escolha do banco de dados que mais se adequa ao projeto desta dissertação.

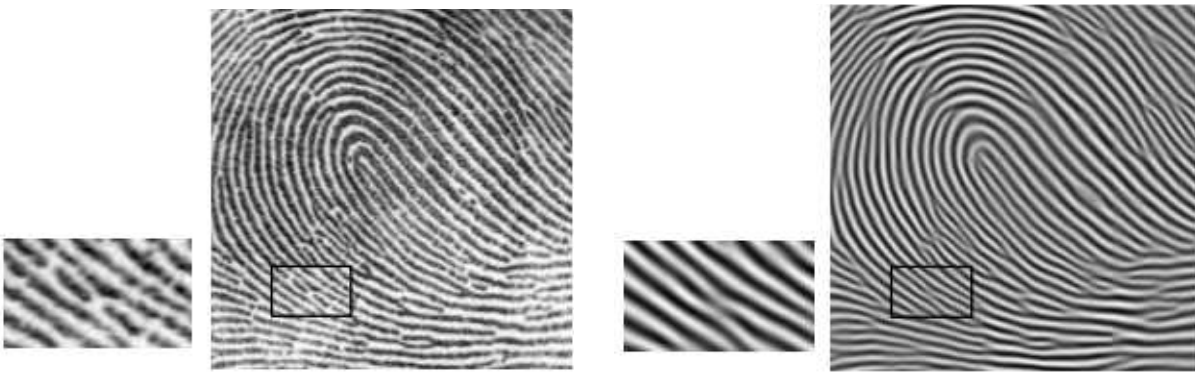
Durante a etapa da representação, a imagem capturada é preparada para ser armazenada. O tamanho, resolução (DPI - *Dots Per Inch*), algoritmos de compressão deverão ser definidos para que todas as imagens estejam de acordo o mesmo padrão. Alguns algoritmos, que não usam minúcias, representam a imagem de uma forma tal como frequências das linhas da impressão (BRADLEY; BRISLAWN; HOPPER, 1993) que facilitam a verificação posterior.

3.1.2 Pré-processamento

As imagens adquiridas raramente possuem uma qualidade ideal por diversas razões, tais como ruido, tinta borrada, entre outros. Portanto, as imagens precisam ser recuperadas e transformadas em imagens mais nítidas. Uma série de filtros são aplicados às imagens com esse objetivo. Esta etapa é a de maior importância e dificuldade pois a qualidade das minúcias extraídas será definida. Se o conjunto de minúcias não for extraído corretamente, certamente influenciará negativamente o resultado do processo de verificação das impressões.

Uma das sequências de transformações e filtros utilizados durante o pré-processamento consiste de normalização, estimativa da orientação, estimativa da frequência das linhas, filtro de Gabor, segmentação, binarização e afinamento (THAI, 2003).

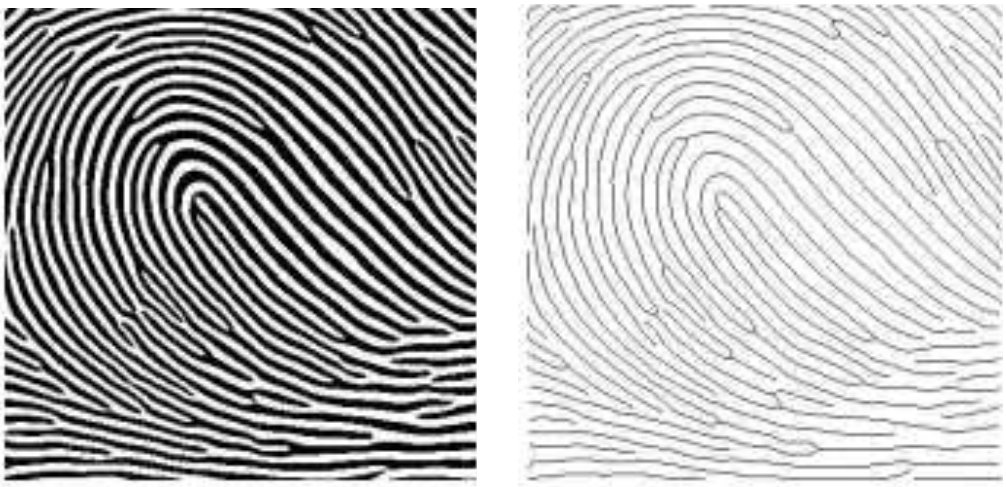
A Figura 9a mostra a imagem original imediatamente após sua aquisição, e a Figura 9b mostra a imagem após a aplicação dos quatro primeiros processos, ou seja, a



(a) *Antes da filtragem* (b) *Após a filtragem*

Figura 9: Aplicação de filtros na imagem da impressão digital

normalização, estimativa da orientação, estimativa da frequência das linhas e o filtro de Gabor.



(a) *Após binarização* (b) *Após afinamento*

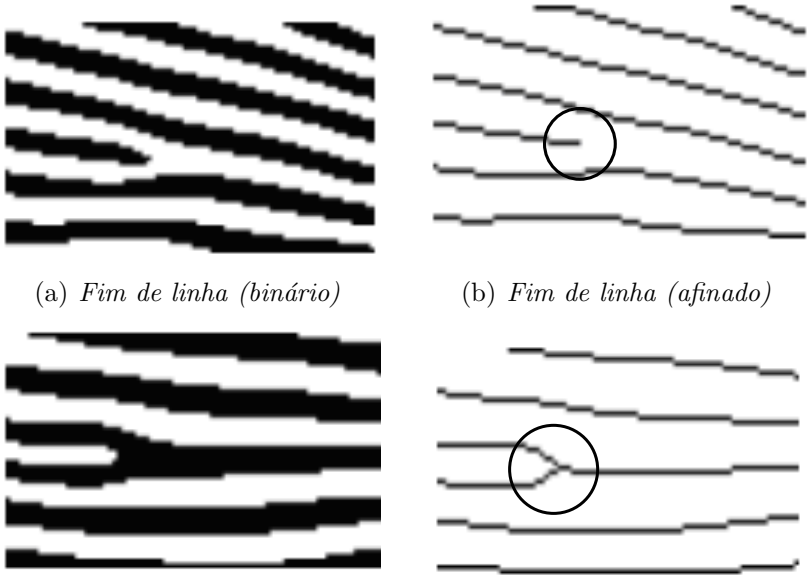
Figura 10: Aplicação de mais filtros durante o pré-processamento

A Figura 10a mostra a imagem após a binarização e a Figura 10b mostra a imagem após o afinamento. Finalizado o pré-processamento, a extração das minúcias é realizada a partir da imagem resultante.

3.1.3 Extração das características

As impressões digitais possuem várias características, tais como os desenhos que as linhas proporcionam, os pontos que parecem ser o centro das curvas entre outros. As minúcias a serem extraídas são apenas de dois tipos: final de linha conforme mostra a Figura 11a

antes do afinamento e Figura 11b após o afinamento, e bifurcação de linha como está ilustrado na Figura 11c antes do afinamento e Figura 11d após o afinamento.



(a) *Fim de linha (binário)* (b) *Fim de linha (afinado)*

(c) *Bifurcação (binário)* (d) *Bifurcação (afinado)*

Figura 11: Tipos de minúcias

Além do tipo, as minúcias possuem duas outras características, que estão ilustradas na Figura 12. São a posição (x_0, y_0) em que a minúcia se encontra em relação à origem da imagem e o ângulo a que é dado pela orientação da linha em que a minúcia se encontra. Ao final da extração, a imagem dará origem a uma lista de minúcias que será guardada ou comparada com outra previamente armazenada na etapa da verificação.

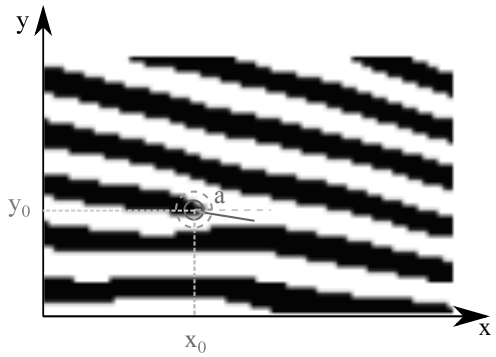


Figura 12: Posição e ângulo das minúcias

Para a realização da extração das minúcias que foram usadas nos testes, foi usado o software NBIS. Trata-se de uma ferramenta desenvolvida pelo NIST (*National Institute of*

Standards and Technology). O resultado da aplicação do NBIS a uma imagem de impressão digital retorna uma lista de minúcias, consistindo em uma localização caracterizada por x e y, ângulo e qualidade. A qualidade é um número que varia de 0 à 100 e é calculada por um pacote da ferramenta NBIS que implementa uma rede neural artificial para tal fim. Minúcias falsas como as da borda da impressão digital tendem a receber um valor mais baixo para a qualidade. O uso do resultado da extração será discutido na Seção 3.5.

3.2 Comparação

O Algoritmo Tolerante à Elasticidade da Pele (SETA - *Skin Elasticity Tolerant Algorithm*) (RATHA et al., 1996) é um algoritmo de verificação de impressão digital que usa um método clássico e bastante conhecido. Este consiste em alinhar as impressões digitais da melhor forma possível antes da avaliação e atribuição de uma nota que quantifica a semelhança das mesmas.

O melhor alinhamento consiste em encontrar as melhores rotação Δ_a e translações Δ_x, Δ_y que devem ser aplicadas em uma das impressões (i.e. conjunto de minúcias) de modo que elas tenham o maior número de características em comum. Outra opção para o alinhamento é a aplicação de uma fator de escala Δ_s em uma da imagens, que pode ser também aplicada nessa etapa porém não será utilizada neste projeto pois as amostras sempre terão uma resolução pré-definida. Essa etapa recebe o nome de registro (*registration*). Para encontrar o melhor alinhamento, é feita a comparação entre todas as minúcias de cada impressão digital, guardando a relação entre a rotação e as translações (Δ_a, Δ_x, Δ_y). No final das comparações, a relação com o maior número de ocorrências de votos será declarada como o melhor alinhamento.

O Algoritmo SETA não registra somente as melhores relações mas também aqueles que dizem respeito à vizinhança. Dessa forma, o algoritmo estará considerando a elasticidade da pele. O Algoritmo 1 apresenta os detalhes do SETA, onde $MA = \{ma_1, ma_2, ma_3, ..., ma_{na}\}$ é o conjunto de minúcias armazenado e na é o número total de minúcias armazenadas e $ME = \{me_1, me_2, me_3, ..., me_{ne}\}$ é o conjunto das minúcias de entrada e ne é o número total de minúcias de entrada enquanto m_{rot} é uma minúcia temporária que armazena o resultado de uma minúcia rotacionada. $E_a, E_x e E_y$ representando a tolerância usada para a elasticidade para o ângulo, x e y, respectivamente.

Apesar de eficiente, o algoritmo não está adequado a implementação em cartão pois

Algoritmo 1 Algoritmo SETA

entrada ME, MA
saída (Δ_a, Δ_x, Δ_y) mais votado

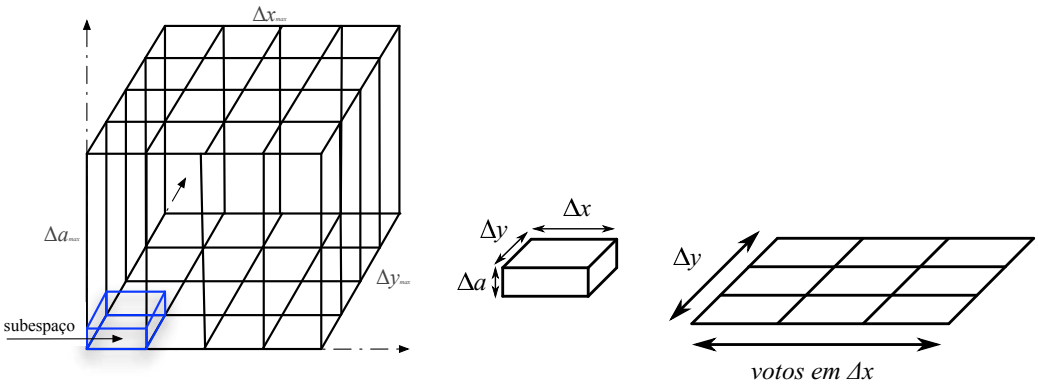
para cada $me_i \in ME$ **faça**
  para cada $ma_j \in MA$ **faça**
    $\delta_a = ma_j.a - me_i.a$
    Rotacione me_i no ângulo δ_a e armazene o resultado em m_{rot}
    $\delta_x = m_{rot}.x - ma_j.x$
    $\delta_y = m_{rot}.y - ma_j.y$
    para $\Delta_a := (\delta_a - E_a) \rightarrow (\delta_a - E_a)$ **faça**
      para $\Delta_x := (\delta_x - E_x) \rightarrow (\delta_x - E_x)$ **faça**
        para $\Delta_y := (\delta_y - E_y) \rightarrow (\delta_y - E_y)$ **faça**
          Incremente (Δ_a, Δ_x, Δ_y) no acumulador de votos
          Salve (Δ_a, Δ_x, Δ_y) se for o mais votado
        fim para;
      fim para;
    fim para;
  fim para;
fim para;

exige muita memória de armazenamento dinâmico para guardar os contadores de todas as possibilidades de rotações e translações. Algumas otimizações foram propostas afim de torná-lo mais adequado ao processamento em cartões inteligentes. Tais otimizações foram inspiradas em (CHOUTA et al., 2012), que trata da implementação do algoritmo SETA em um co-processador de *hardware*, visando um menor uso de memória e tempo de processamento. Estas adequações são explicadas no restante desta seção.

3.2.1 Organização em subespaços

Para remediar o problema do volume de armazenamento necessário para guardar as translações e rotações, foi usada a estratégia de dividir o espaço total de busca em volumes menores, chamados subespaços. A cada iteração, são consideradas apenas as translações e rotações pertencentes ao subespaço considerado em cada instante. A Figura 13a ilustra essa estratégia de subdivisão do espaço de busca. O espaço total de busca pode ter dimensões grandes mas o subespaço estudado, a cada iteração, pode ser dimensionado de acordo com a memória disponível.

A Figura 13b mostra dois subespaços com diferentes alturas mas que requisitam o mesmo tamanho de memória. A altura, no caso, indica a variação do ângulo, ou seja, em um mesmo subespaço serão considerados apenas votos com a mesma diferença angular.



(a) *Espaço de busca* (b) *Subespaço usado*

Figura 13: Organização em subespaços

Para a implementação, foi escolhido o valor da altura de cada subespaço como unitário, ou seja, serão averiguadas partes das lâminas do volume total. O tamanho do subespaço e o número de subespaços estudados serão determinados pelos limites de translação e rotação adotados como será visto na Seção 3.5.

3.2.2 Tabela de acesso

Para melhorar o tempo de busca das minúcias armazenadas a serem utilizadas a cada iteração do algoritmo foi proposta uma tabela que irá armazenar os índices de um outra tabela que contém as minúcias, ou seja, na tabela de acesso estará o índice inicial e final das minúcias que possuem o mesmo ângulo. A estrutura destas duas tabela é ilustrada na Figura 14, onde $Angulo_{max}$ representa o maior ângulo possível na representação escolhida. Essa tabela é usada para facilitar a procura dos ângulos correspondentes à diferença de ângulo estudada em um subespaço específico de uma iteração.

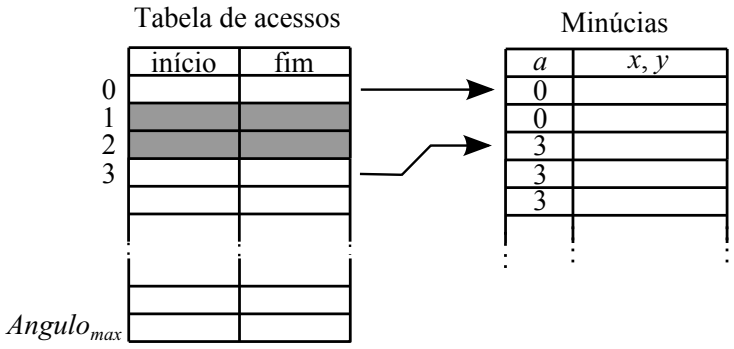


Figura 14: Tabela de acessos e minúcias

Para a implementação no cartão inteligente, é necessário organizar a tabela na mesma função em que as minúcias são armazenadas. Dessa forma, fica garantida a coerência dos dados armazenados (Minúcias e Tabela de acesso).

3.2.3 Algoritmo baseado em subespaços

Feitas as alterações propostas, foi desenvolvida uma nova versão do SETA, conforme descrito no Algoritmo 2. Este último testa os valores para cada rotação possível em cada subespaço existente e guarda as melhores relações com suas vizinhanças, sem que sejam ultrapassados os limites do subespaço. Considere TAM_{SUB} o número total de subespaços utilizados, α o limite considerado de diferença absoluta de ângulo entre duas minúcias, T_{inicio} e T_{fim} o início e fim, respectivamente, da área que terá os votos incrementados obedecendo o limite do subespaço da iteração.

Algoritmo 2 Algoritmo SETA melhorado para execução em smart card

entrada ME, MA, α
saída (Δ_a, Δ_x, Δ_y) mais votado

para $\Delta_a := -\alpha \rightarrow \alpha$ **faça**
  para $Sub_s, s := 0 \rightarrow TAM_{SUB} - 1$ **faça**
    para cada $me_i \in ME$ **faça**
      busque $ma_j \in MA$ utilizando a tabela de acesso
      $\delta_a = ma_j.a - me_i.a$
      Rotacione me_i no ângulo δ_a e armazene o resultado em m_{rot}
      $\delta_x = m_{rot}.x - ma_j.x$
      $\delta_y = m_{rot}.y - ma_j.y$
      $T_{inicio} = (\delta_x, \delta_y) - (E_x, E_y)$
      $T_{fim} = (\delta_x, \delta_y) + (E_x, E_y)$
      para $\Delta_x := T_{inicio}.x \rightarrow T_{fim}.x$ **faça**
        para $\Delta_y := T_{inicio}.y \rightarrow T_{fim}.y$ **faça**
          Incremente (Δ_x, Δ_y) no acumulador de votos
          Salve (Δ_a, Δ_x, Δ_y) se for o mais votado
        fim para;
      fim para;
    fim para;
  fim para;
  Limpe o acumulador de votos
fim para;

O algoritmo foi implementado na linguagem Java usando a plataforma Java Card e simulado usando as ferramentas descritas na Seção 1.1.3. Os detalhes da implementação e os resultados obtidos serão descritos e analisados na Seção 3.5.

O resultado do novo algoritmo SETA (Algoritmo 2) é a melhor translação e rotação. Note que a comparação ainda precisa ser feita. O conjunto de translação e rotação escolhido na fase de registro é então aplicado às minúcias de entrada para então serem comparadas com as minúcias armazenadas. O tipo de comparação mais simples é a imposição de limites em torno das minúcias armazenadas, ou seja, se a diferença entre uma minúcia de entrada e uma armazenada tiver Δ_a, Δ_x e Δ_y menores do que certos limites pré-definidos, as minúcias serão consideradas autênticas. A decisão final é tomada baseando-se no número total de minúcias declaradas como autênticas. Apesar de muito simples, esse tipo de comparação não leva em conta a elasticidade da pele.

O trabalho estudado (CHOUTA et al., 2012) sugere um sistema de notas computando a distância entre as minúcias, através do uso de uma distribuição Gaussiana pré-definida. Menores distâncias recebem uma nota alta e, a partir de uma certa distância, a nota passa a ser baixa. Uma medição semelhante foi utilizada na implementação deste projeto mas, para simplificar o processamento, foi utilizada uma distribuição linear para relacionar a nota com a distância. Maiores detalhes são expostos na Seção 3.5.

3.3 Banco de Dados

É necessário definir como ou onde serão obtidas as amostras que serão usadas nos testes. Novamente, por ser a biometria mais utilizada atualmente, existem muitos bancos de dados de amostras de impressões digitais (MFCP2, 2004), (MAIO et al., 2002a), (MAIO et al., 2002b), (MAIO et al., 2004). A escolha do banco de dados ajudará na escolha do número de subespaços utilizados e na diferença de ângulo estudada.

O banco de dados escolhido foi o banco usado pela FVC 2000 (*Fingerprint Verification Competition*) por ser utilizado em muitos trabalhos publicados e possuir amostras de impressões digitais capturadas por sensores em ambientes controlados, assim como são as verificações em cartões inteligentes.

O FVC 2000 possui quatro bancos diferentes, cada um com uma forma diferente de captura das impressões digitais. Foi escolhido o banco de dados que usa o sensor ótico de baixo custo. Esse banco possui 8 amostras de impressões de 110 diferentes dedos, capturadas em imagens de 300×300 pixels com resolução de 500 dpi. A Figura 15 mostra alguns exemplos do banco de dados utilizado, note que as imagens possuem diferentes características, tais como rotação, problemas de pele, linhas imprecisas.

Figura 15: Exemplos de amostras do banco de dados FVC 2000

Informações adicionais sobre a parametrização da captura podem ser obtidas em (MAIO et al., 2002a). As principais serão citadas a seguir.

- A maioria da impressões pertencem à alunos de 20 à 30 anos (aproximadamente 50% masculino).
- Até quatro diferentes impressões foram coletadas de cada voluntário.
- Os voluntários não receberam treinamento e e as amostras foram coletadas sem qualquer preocupação com a qualidade.
- Os aparelhos não foram sistematicamente limpos durante as aquisições.
- A rotação relativa máxima entre as amostras de uma mesma impressão é de -15º à 15º assim como é garantido que existe uma área em comum entre elas.

3.4 Implementação

O primeiro passo durante a implementação do Algoritmo 2 consiste em definir as variáveis usadas para cada uma das estruturas utilizadas, tais como a tabela de acesso e tabela de

minúcias e para as mensagens de transmissão. É necessário definir como serão armazenadas as minúcias e o número máximo de minúcias. A configuração do tamanho do subespaço e número de subespaços estudados entre outras configurações são abordadas ao longo desta seção.

Como visto na Seção 3.1, o resultado da ferramenta de extração utilizada (NBIS) apresenta cada minúcia com uma posição (x, y), um ângulo e uma medida de qualidade. Para o banco de dados utilizado, x e y podem variar de 0 à 299, logo, foi escolhida a variável short para seu armazenamento uma vez que o *byte* pode conter apenas valores de -128 à 127. O resultado da rotação da minúcia dado pelo software de extração varia de 0 à 63, logo, a variável *byte* já se mostra adequada para seu armazenamento. Totalizando assim em 5 *bytes* o tamanho de memória necessário para armazenar cada minúcia. É necessário também escolher o número mínimo de minúcias por impressão digital antes de proceder com o armazenamento ou a comparação visto que, levando em consideração a fase de registro (alinhamento), impressões contendo apenas uma minúcia sempre teriam resultado de 100% de proximidade. Em casos de criminalística, normalmente, são necessárias 12 minúcias (*the 12-point guideline*) muito semelhantes para que duas impressões sejam consideradas da mesma pessoa (MAIO; JAIN, 2009). Obviamente, sistemas biométricos não necessitam tamanha precisão, mas essa informação é usada como base para estabelecer o número mínimo de minúcias aceitas como 10. Isso significa que qualquer impressão contendo menos de 10 minúcias não poderá ser usada para comparações e o cartão sempre retornará um código de erro.

O número máximo de minúcias foi escolhido levando em conta o número de minúcias de alta qualidade ($> 60\%$) extraídas, que variou entre 5 e 33 minúcias para as amostras de impressão digital selecionadas. Também foi levado em consideração a capacidade máxima de uma transmissão do Java Card, que não pode ultrapassar 128 *bytes*. Como cada minúcia ocupa 5 *bytes*, foi escolhido um máximo de 25 minúcias, totalizando 125 *bytes*. Foi definido que as minúcias devem ser sempre transmitidas em ordem crescente de ângulo para facilitar a construção da Tabela de Acesso quando uma impressão é armazenada.

A Tabela de Acesso deve conter os índices de início e fim das minúcias para cada ângulo possível (0 à 63). Logo, esta tabela é armazenada em 2 vetores de 64 posições, um para alocar o índice da primeira minúcia na tabela de minúcias que possui o ângulo da

posição do vetor e outro para alocar o índice da última.

O tipo de subespaço escolhido considera apenas uma rotação por vez (conforme explicado na Seção 3.2.1). O trabalho usado como base para esta implementação (CHOUTA et al., 2012) conclui que quanto maior a memória utilizada para o subespaço, menor será o tempo de processamento. A RAM de um cartão inteligente normalmente tem em torno de 3kb. Foi escolhido um subespaço de tamanho 32×32 *bytes* utilizando 1kb (1/3 do tamanho total). Dessa forma, a alocação do subespaço não irá interferir na alocação de outras variáveis desta e de outras aplicações.

O Algoritmo 2 foi implementado em Java Card mas, por ser uma plataforma com várias restrições, algumas modificações foram necessárias. Na linha 7, é feita a operação de rotação de um ponto em um certo ângulo. Essa operação, ilustrada na Equação 1, teve de ser implementada usando apenas variáveis inteiras pois o Java Card não possui variáveis com números decimais nem funções matemáticas.

$$\begin{aligned} x_{rot} &= xcos(\theta) - ysen(\theta) \\ y_{rot} &= xsen(\theta) + ycos(\theta) \end{aligned} \tag{1}$$

Para tal, foram implementadas duas tabelas de referência com os valores de seno e cosseno para cada ângulo possível de ser usado. Para contornar o problema de precisão, os valores de seno e cosseno foram multiplicados por 100. Para o cálculo de $xsen(\theta)$, por exemplo, o x é multiplicado pelo valor de $sen(\theta)$ lido na tabela, que está multiplicado por 100, e depois é dividido por 100 para dar o resultado correta sem a necessidade de usar casas decimais. Note que um valor maior do que 100 não poderia ser usado para aumentar a precisão pois existe o limite positivo da variável short é de 32767. Usar a multiplicação por 100 limita os valores de x e y a 320. Se fosse usado o valor de 1000, o máximo seria de apenas 32.

Como apontado na Seção 3.2, após a fase de registro, que permite escolher da melhor rotação e translação, é necessário comparar as minúcias usando-as. Para as comparações entre todas as minúcias armazenadas, i.e. $na = 25$, com todas as minúcias de entrada, i.e. $ne = 25$, são atribuídas notas. Para armazenar as notas obtidas, foi utilizado o mesmo espaço de memória usado anteriormente (na fase de registro) para guardar os votos do subespaço (32×32 *bytes*). Como o número máximo de minúcias é 25, a memória máxima necessária é de 25×25 *bytes* garantindo a adequação do espaço. O espaço usado para guardar as notas será referido como *matriz*.

Algoritmo 3 Algoritmo do cálculo de proximidade entre minúcias

entrada (ME, MA), Δ_a, Δ_x e Δ_y mais votados na fase de registro
saída $matriz$

1: **para cada** $me_i \in ME$ **faça**
2: Rotacione me_i o ângulo Δ_a e armazene o resultado em m_{rot}
3: Traslade m_{rot} em Δ_x e Δ_y
4: **para cada** $ma_j \in MA$ **faça**
5: $\delta_x = m_{rot}.x - ma_j.x$
6: $\delta_y = m_{rot}.y - ma_j.y$
7: $\delta_a = m_{rot}.a - ma_j.a$
8: $pTrans = 2(50 - (|\delta_x| + |\delta_y|))$
9: $pRot = 4 - |\delta_a|$
10: **se** $pTrans \leq 0$ **OU** $pRot \leq 0$ **então**
11: $proximidadeMinucia = 0$
12: **senão**
13: $proximidadeMinucia = (pTrans * pRot)/4$
14: **fim se**
15: $matriz[i, j] = proximidadeMinucia$
16: **fim para**
17: **fim para**

O Algoritmo 3 detalha a forma de pontuação de cada comparação entre as minúcias armazenadas e de entrada. A variável $pTrans$ representa a *proximidade* em termo de distância, note que a distância é calculada de forma simplificada, apenas somando os módulos da diferença devido ao baixo poder computacional dos cartões. Essa variável denota uma proximidade máxima de 100 caso $|\delta_x| + |\delta_y| = 0$. De forma semelhante, na variável $pRot$ será armazenada a proximidade do ângulo entre as minúcias. $pRot$ alcança o valor máximo de 4 no caso de $\delta_a = 0$ e terá zero como seu valor atribuído no caso de $\delta_a \geq 4$. Caso sejam positivas, $pTrans$ e $pRot$ serão multiplicadas e o resultado dividido por 4 e, dessa forma, resultará em uma nota de comparação, que pode variar de 0 à 100, entre duas minúcias. Essa nota é então armazenada na matriz de resultados. Vale ressaltar que quanto maior a distância, menor a nota atribuída.

Ao final da execução do algoritmo, a matriz de resultados estará preenchida com os resultados de cada uma das comparações entre as minúcias dos conjuntos de minúcias armazenadas (MA) e de entrada (ME). São então somados os melhores resultados sem que haja repetição de minúcias cujos resultados já foram considerados, ou seja, o resultado da comparação entre me_3 e ma_4 não pode ser considerado se o resultado da comparação me_3 e ma_1 já foi usado pois estaria repetindo a minúcia me_3. O maior resultado da $matriz$ (100) é desprezado pois o sistema de votos da fase de registro garante que ao menos uma

minúcia consiga um valor próximo a 100 e isso resulta em uma falsa proximidade para comparações falsas. O resultado final da soma dos maiores resultados da *matriz* é então divido pelo número total de resultados considerados para obter uma média. Como nenhuma minúcia pode ter 2 resultados, o total de resultados considerados será exatamente o total de minúcias do menor conjunto subtraído de 1 por conta da exclusão do melhor resultado. O resultado final da comparação pode variar de 0 à 100. Este resultado final é referenciado como a *proximidade* entre duas impressões ou conjunto de minúcias.

O último passo da implementação é a definição da variação de translação máxima e rotação máxima que delimitarão o espaço total de busca do algoritmo. Essa escolha afetará diretamente o tempo de execução do algoritmo e a qualidade das comparações. Alguns testes foram feitos afim de averiguar a relação entre o resultado e o tempo usando o banco de dados que ilustra situações semelhantes ao uso real de um equipamento de captação de impressões digitais.

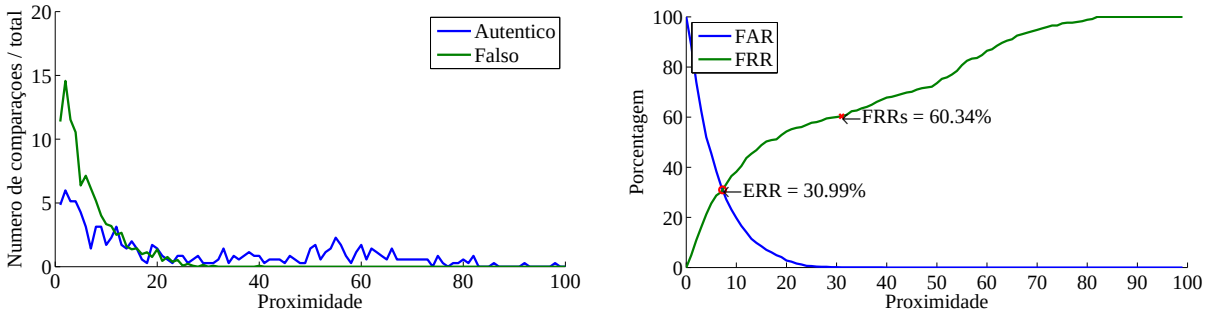
3.5 Resultados

Para a avaliação da implementação no cartão inteligente foram usadas 8 repetições de 10 diferentes impressões do banco de dados introduzido na Seção 3.3. Logo, existem 6400 comparações possíveis no total. Se consideramos apenas as repetições de uma mesma impressão digital, teremos 64 comparações possíveis e é esperado que essas comparações atinjam os maiores valores de proximidade. Essas comparações serão referenciadas como comparações autênticas. Como foram usadas 10 impressões diferentes, serão 640 as comparações autenticas do total de 6400 comparações. As comparações entre as repetições de diferentes impressões serão referenciadas como comparações falsas e somam um total de 5760.

Considere duas impressões A e B. Note que a contagem de comparações possíveis é feita considerando tanto a comparação de A com B quanto a de B com A, ou seja, quando A está armazenada no cartão e B tenta o acesso e quando ocorre o contrário. Ambos resultados serão considerados no teste pois existe a possibilidade que eles sejam diferentes. Isso ocorre devido ao uso de inteiros para realizar as funções *seno* e *cosseno*, como explicado anteriormente na Seção 3.4. As rotações e translações acabam apresentando diferentes resultados e, com isso, os votos feitos na fase de registro podem ser diferentes e isso influenciará o resultado final obtendo diferentes valores de proximidade.

Como abordado anteriormente, foi escolhido um subespaço de tamanho 32×32 que guarda as diferenças em x e y para duas minúcias comparadas. Logo, o limite da translação deve ser múltiplo de 32. Primeiramente, foi escolhido usar o limite de translação de -32 à 31 à um passo de 32 em x e y. Ou seja, é necessário usar o subespaço em 4 iterações para conseguir testar todas diferenças de x e y possíveis. A diferença de ângulo usada foi de -1 à 1 (o ciclo do ângulo variando de 0 à 64) por ser as menores rotações possíveis.

Os resultados obtidos serão apresentados usando 4 tipos de gráficos. Estes serão explicados com maiores detalhes pois serão usados para analisar os resultados dos testes de todas as biometrias implementadas no projeto desta dissertação. As comparações entre uma amostra e ela mesma, ou seja, exatamente o mesmo conjunto de minúcias serão consideradas apenas nos gráfico que envolvem tempo pois sempre terão nota de 100, que representa uma situação praticamente impossível.



(a) *Distribuição dos resultados* (b) *FAR e FRR*

Figura 16: Resultados utilizando 4 subespaços e ângulo $a \in [-1, 1]$

A Figura 16a mostra os resultados, que são dados pela proximidade entre duas impressões para as comparações autenticas e falsas encontrados nos testes. Ou seja, o gráfico mostra quantas comparações obtiveram uma certa proximidade. Para que ambos os tipos de comparação pudessem ser representados em um mesmo gráfico, foi considerado o percentual em relação ao total (lembrando que são possíveis 5760 comparações falsas e apenas 640 comparações autenticas). Este tipo de gráfico será referenciado como a *distribuição* dos resultados. Figura 16a é possível notar que há uma grande concentração dos resultados das comparações falsas nas proximidades menores que 30. Já as comparações autênticas encontram-se distribuídas em valores de 0 à 80. Em um sistema biométrico, é necessário que haja um limite para a proximidade que separa os resultados que conseguirão o acesso dos que serão recusados. Note que, intuitivamente, parece interessante

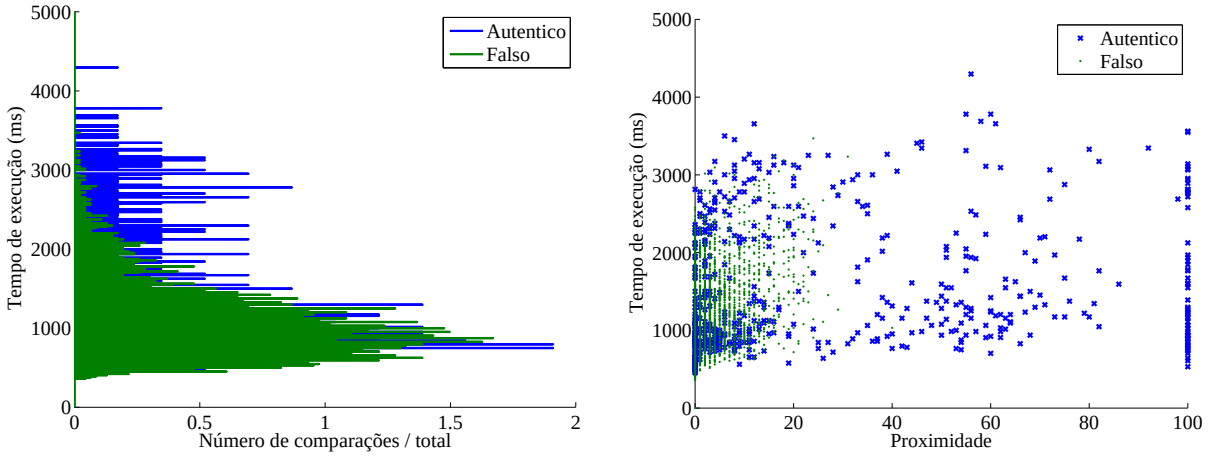
escolher um limite de aceitação em torno da proximidade 30 pois não garantiria acesso à maioria das comparações autenticas , mas recusaria a maioria das considerações falsas.

Para conseguir conclusões mais exatas a cerca do limite de aceitação, serão usadas duas taxas introduzidas no Capítulo 1. Primeiramente, o FRR (*False Rejection Rate*), a taxa de falsas rejeições, é a porcentagem de comparações autenticas que não alcançaram o limite de aceitação e foram consideradas falsas.. A taxa de falsas aceitações é o FAR (*False Acceptance Rate*) que avalia a situação oposta, ou seja, é a taxa de comparações falsas que ultrapassaram o limite de aceitação e foram consideradas autênticas. A Figura 16b mostra o gráfico que relaciona FAR e FRR para diferentes valores de proximidade considerados como limite aceitação. Note que para um limite de aceitação de proximidade igual a 100 nenhuma comparação é aceita como verdadeira, logo, o FRR é de 100% e o FAR (taxa de falsas aceitações) é de 0%, pois todas as comparações falsas foram realmente consideradas como falsas. Por outro lado, escolhendo o limite de aceitação como uma proximidade de 0 o inverso ocorre.

Dois pontos são importantes na Figura 16b. O ponto, onde as porcentagens de erro são iguais, é chamado de ERR (*Equal Error Rate*) e serve como indicativo de qualidade do algoritmo de comparação. O ideal é possuir o menor ERR possível ou até mesmo zero. O outro ponto importante é o FRR seguro, que é o valor de FRR para o mesmo limite de aceitação em que o FAR <0,1%. Esse ponto é importante pois mostra a taxa de falsas rejeições em que a o falso aceite ocorre apenas 1 vez a cada 1000 falsas comparações. Para um sistema de segurança é importante que impostores não consigam o acesso. No caso em análise, o ERR foi de 30,99% enquanto o FRR seguro foi de aproximadamente 60%. Isso indica que o usuário, dono do cartão, normalmente tentaria mais de uma vez para conseguir o acesso mas teria a garantia que um outro indivíduo praticamente nunca conseguiria acesso.

Outro aspecto que não pode ser ignorado é o tempo de execução. Um sistema de autenticação usando cartões inteligentes não pode ter um tempo de execução muito grande pois normalmente é feito para atender a eventos em tempo real. A Figura 17a mostra os tempos de execução (considerando o tempo de transmissão das minúcias) das comparações autenticas e falsas. O gráfico mostra quantas comparações obtiveram um determinado tempo de execução. A exemplo do gráfico de distribuição dos resultados, foi considerado o percentual em relação ao total para que os tempos das comparações

autenticas e falsas pudessem ser representados em um mesmo gráfico. Nota-se que grande parte das comparações obtiveram tempo de execução em torno de 1 segundo mas algumas comparações chegaram a demorar perto de 4 segundos.



(a) *Tempo de execução* (b) *Proximidade em relação ao tempo*

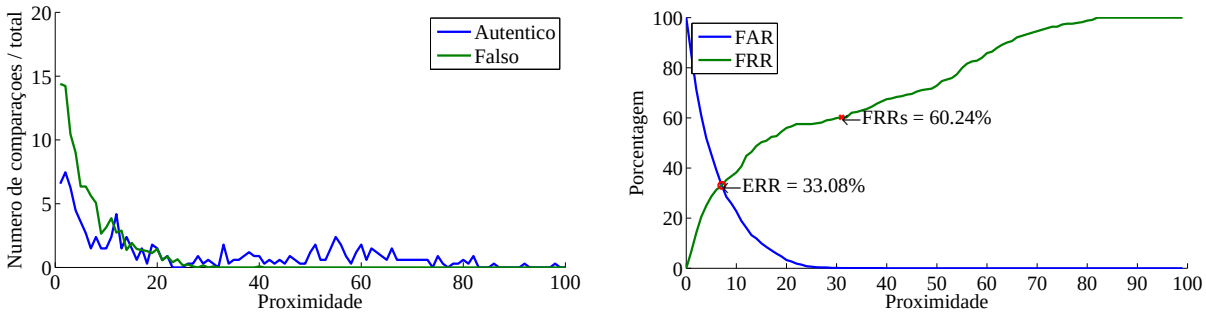
Figura 17: Tempo de execução utilizando 4 subespaços e ângulo $a \in [-1, 1]$

A Figura 17b mostra o gráfico que relaciona o resultado de uma comparação com seu tempo de execução, ou seja, cada ponto no gráfico representa uma comparação entre duas impressões. É possível notar que não existe uma relação entre o tempo de execução e o resultado da comparação. Isso ocorre pois as comparações entre as minúcias para a fase do registro e a comparação final ocorre da mesma forma independente das distâncias entre elas.

Apesar de apresentar valores de tempo de execução aceitáveis, os resultados da autenticação foram muito falhos. Isso de certa forma, é esperado pois foram escolhidos valores baixos para a diferença de translação e de rotação. Para avaliar a importância desses valores, o próximo teste proposto considera o aumento da variação da rotação estudado para -5 à 5. A Figura 18 mostra os gráficos da distribuição dos resultados e relação de FRR e FAR em termo de proximidade para este novo teste.

O resultado do novo teste indica que as comparações falsas conseguiram notas maiores pois a porcentagem igual de erro (ERR) aumentou consideravelmente. Já o FRR seguro permaneceu praticamente inalterado, indicando que a variação do ângulo não é de extrema importância para melhorar os resultados.

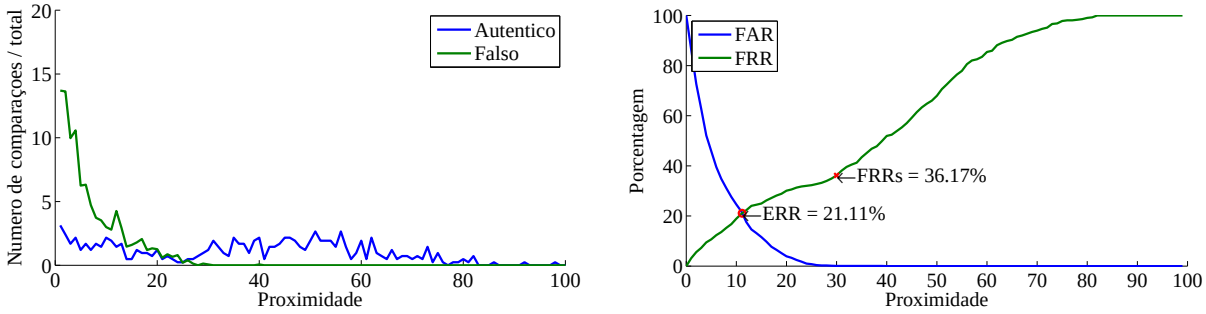
Um novo teste foi proposto como uma nova tentativa de melhorar os resultados. Desta vez, o aumento foi na variação da translação enquanto a variação do ângulo con-



(a) *Distribuição dos resultados* (b) *FAR e FRR*

Figura 18: Resultados utilizando 4 subespaços e ângulo $a \in [-5, 5]$

siderada voltou a ser de -1 à 1. Utilizando o mesmo tamanho de subespaço (32×32), o espaço de busca foi aumentado para considerar translações de -64 à 63 em x e em y (totalizando $4 \times 4 = 16$ subespaços analisados). Os resultados dos testes são mostrados nos gráficos da Figura 19 .

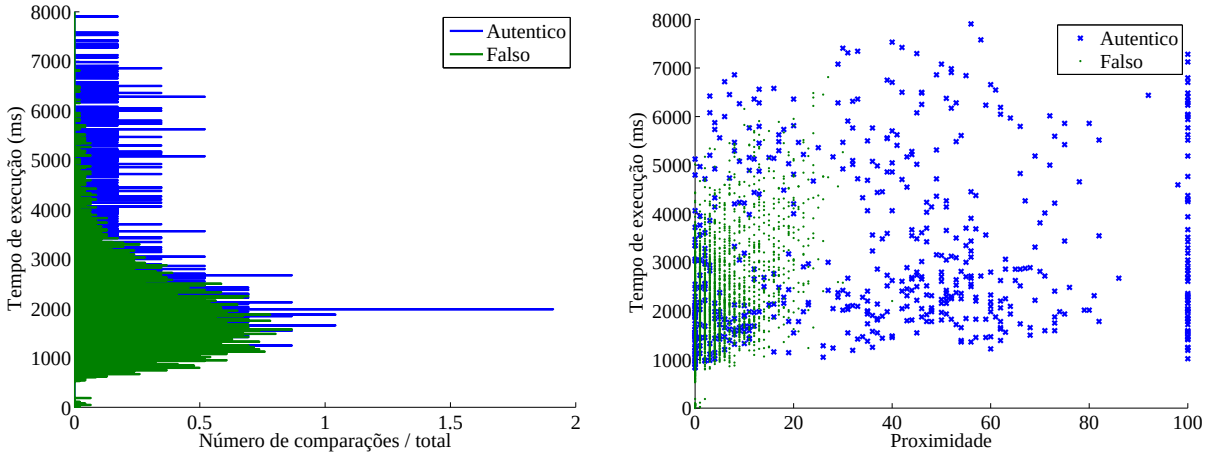


(a) *Distribuição dos resultados* (b) *FAR e FRR*

Figura 19: Resultados utilizando 16 subespaços e ângulo $a \in [-1, 1]$

O gráfico da Figura 19b que relaciona FAR e FRR agora traz resultados mais atraentes. A taxa de erro igual (ERR) melhorou de 30,99% para 21,11% e o FRR seguro sofreu uma grande queda, ficando agora em torno de 36%. O resultado pode ser entendido de forma prática como 1 erro a cada 3 tentativas de comparações Autênticas tornando o resultado mais próximo de uma realidade aceitável contra, praticamente 2 erros a cada 3 tentativas apresentado no primeiro teste. Certamente essa melhoria nos resultados também trará consequências ao tempo de execução da comparação.

O gráfico da Figura 20a mostra que a maioria das comparações teve seu tempo de execução entre 0,5 e 3,5 segundos mas alguma comparações demoraram a quase 8 segundos. Afim de estudar mais a fundo a relação entre as variações de rotação e trans-



(a) *Tempo de execução* (b) *Proximidade em relação ao tempo*

Figura 20: Tempo de execução utilizando 16 subespaços e ângulo $a \in [-1, 1]$

lação, foram feitos mais testes utilizando mais variações. Os resultados desses testes são mostrado na Tabela 3.

Tabela 3: Resultado das comparações com diferentes faixas de translações e rotação

Translação	Rotação	Subesp.	ERR	FRR seg.	Tempo médio	Desvio padrão
−32 à 31	−1 à 1	4	30,99	60,34	1371,95	918,16
−32 à 31	−3 à 3	4	30,36	58,39	1895,21	1284,27
−32 à 31	−5 à 5	4	33,08	60,24	2296,75	1534,94
−64 à 63	−1 à 1	16	21,11	36,17	2849,38	1901,37
−64 à 63	−3 à 3	16	20,28	35,40	4876,06	3294,97
−64 à 63	−5 à 5	16	20,55	34,83	6433,86	4258,85
−96 à 95	−1 à 1	36	15,34	31,03	4966,29	3270,78
−96 à 95	−3 à 3	36	16,31	31,25	9274,38	6173,16
−96 à 95	−5 à 5	36	16,14	30,77	12748,45	8291,08

Os resultados indicam que o aumento da faixa de translação analisada, até certo ponto, resulta em grandes melhorias no resultado final do teste, porém paga-se um alto preço com o aumento do tempo de execução. A variação da faixa de rotação estudada associou pouca melhoria em relações ao resultado mas um aumento significativo em tempo de execução. Note que, em alguns casos, o aumento da faixa de rotação estudada ocasionou em um aumento no FRR seguro. Inicialmente essa informação não parece correta, mas o FRR seguro depende também do FAR ser menor que 0,1%, indicando que houve aumento do FAR e isso resultou em anular a diminuição do FRR.

Para um sistema de segurança baseado no uso de cartões inteligentes, os valores mais importantes são o tempo médio, desvio padrão e taxa de FRR seguro. O teste

utilizando diferenças de translação de -64 à 63 e -1 à 1 para rotação proporciona a melhor relação entre resultado de comparação e tempo de execução, apresentando valores aceitáveis em ambos. O limite de aceitação, ou seja, o valor da proximidade que deve ser usado para diferenciar o indivíduo certo do errado deve ser definido com o mesmo valor de proximidade onde ocorre o FRR seguro. Para o caso estudado esse valor é de 30% de proximidade como pode ser visto na Figura 19b.

3.6 Considerações Finais

Este capítulo abordou todos os aspectos da implementação da biometria da impressão digital em um cartão inteligente. O algoritmo SETA foi implementado utilizando algumas melhorias para sistemas embarcados que permitiram um menor uso de memória. O algoritmo se mostrou extremamente dependente das translações e rotações na fase de aquisição. O número de minúcias extraídas e a qualidade da extração também podem alterar o resultado de forma definitiva. Apesar de não apresentar alta porcentagem de acerto, foi demonstrado que o algoritmo pode ser usado em um sistema real. No Capítulo 6, são analisados os resultados das diferentes biometrias implementadas em cartões inteligentes.

Capítulo 4

IMPRESSÃO DA PALMA DA MÃO

COMO visto na Seção 2.2, apesar de ser recente o estudo da biometria da impressão da palma da mão, diversos projetos foram desenvolvidos e métodos foram propostos para encontrar a melhor solução ou a mais apropriada. Levando em consideração a maturidade do método, facilidade de extração, baixa necessidade de memória e o baixo poder computacional necessário para efetuar a comparação, o *PalmCode* (ZHANG et al., 2003) foi o método escolhido para implementação no cartão.

O método de extração do código binário, algoritmo de comparação, banco de dados utilizados, implementação da biometria e os resultados obtidos serão abordados nas Seções 4.1, 4.2, 4.3, 4.4 e 4.5, respectivamente.

4.1 Extração

Em todos os tipos de biometrias, a extração é uma das etapas de maior importância pois o resultado obtido afeta diretamente o desempenho das próximas etapas assim como o resultado final da comparação. A Seção 4.1.1 explica com maiores detalhes da delimitação da área de interesse da imagem da impressão da palma da mão e a Seção 4.1.2 detalha o método utilizado para extração do código binário.

4.1.1 Área de interesse

Para um melhor resultado na geração do código binário é importante que a delimitação da região de interesse seja sempre feita da mesma forma em diferentes imagens, independente da qualidade destas. Zhang (ZHANG et al., 2003) elaborou um sistema de coordenada para realizar as medições usando as junções dos dedos como pontos de referência. São cinco

os principais passos, conforme ilustrados na Figura 21, para a delimitação da área de interesse:

Passo 1. Aplicar um filtro passa-baixa, como o Gaussiano, na imagem original. Utilizar um limite de intensidade na imagem resultante para classificar os *pixels* em branco ou preto como mostrado na Figura 21b.

Passo 2. Obter as bordas das junções dos dedos, (F_1x_j, F_1y_j) e (F_2x_j, F_2y_j), utilizando um algoritmo de reconhecimento de bordas, tais como Sobel e Canny, conforme ilustrado na Figura 21c. A junção entre os dedos anelar e médio não é extraída por não ser usado nos próximos passos.

Passo 3. Encontrar a tangente entre as duas junções. Considere (x_1, y_1) e (x_2, y_2) sendo pontos em (F_1x_j, F_1y_j) e (F_2x_j, F_2y_j) respectivamente. Se a linha $(y = mx + c)$ passando por esse dois pontos satisfaz as inequações $F_iy_j \leq mF_ix_j + c$, para todos os valores i e j (Figura 21d), esta linha será a tangente entre as duas junções.

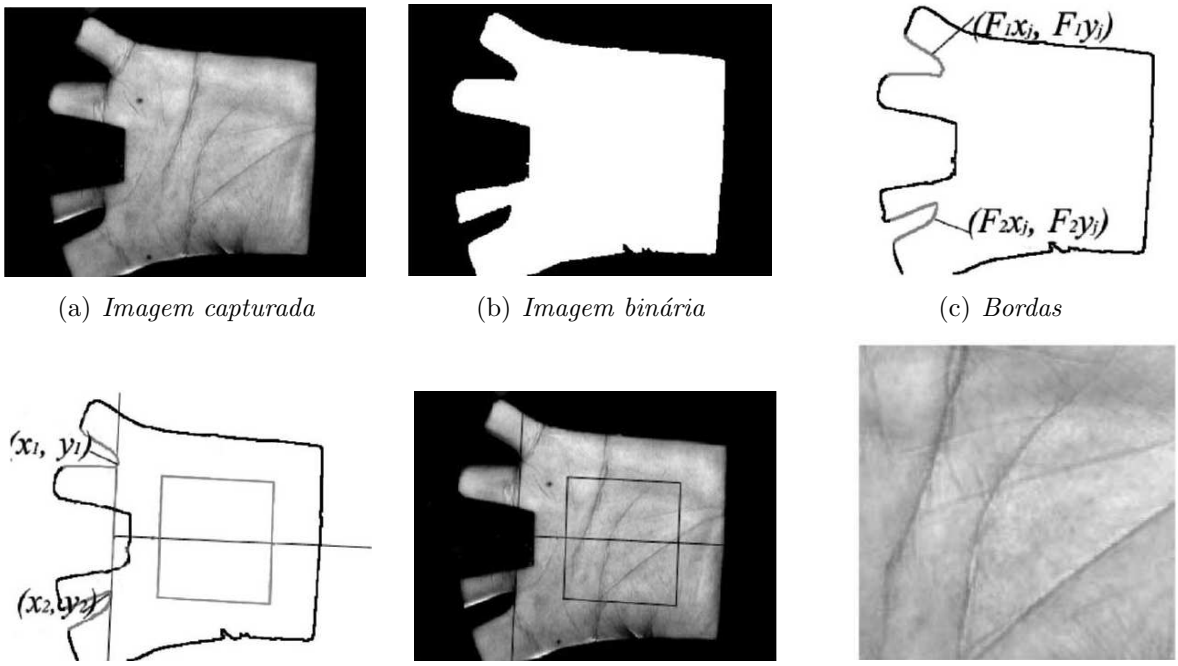
Passo 4. A linha passando entre (x_1, y_1) e (x_2, y_2) será o eixo Y do sistema de coordenadas da palma da mão. A linha perpendicular ao eixo Y que passa pelo ponto médio entre os dois pontos será o eixo X, determinando assim a origem do sistema de coordenadas (Figura 21d)

Passo 5. A área de interesse será uma sub-imagem de tamanho pré-definido baseado no sistema de coordenadas (Figura 21e). Essa sub-imagem será usada para extrair o código binário (Figura 21f).

4.1.2 Código binário da palma da mão

A base para a comparação é a sub-imagem extraída com o auxílio do sistema de coordenadas visto na Seção 4.1.1. A comparação direta da imagem com outra imagem é muito suscetível à luminosidade e a qualidade da imagem capturada. Inspirado no trabalho de Daugman sobre biometria da Iris (DAUGMAN, 1993), Zhang (ZHANG et al., 2003) propôs usar o filtro 2D de Gabor para extração das principais características da impressão da palma da mão. Esse filtro permite neutralizar as diferenças de luminosidade e qualidade, trazendo a possibilidade da comparação direta.

Devido a inexistência de uma ferramenta disponível para extração do *PalmCode* de uma imagem de impressão da palma da mão, foi implementado em MATLAB o filtro 2D de Gabor e o extrator do código. Conforme explicado nas duas próximas seções.



(a) *Imagem capturada* (b) *Imagem binária* (c) *Bordas*

(d) *Sistema de coordenadas* (e) *Delimitação* (f) *Área de interesse*

Figura 21: Sistema de coordenadas para extração da área de interesse

4.1.2.1 Filtro 2D de Gabor

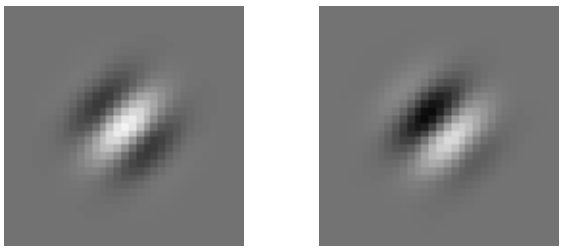
A função 2D de Gabor foi proposta por Daugman (DAUGMAN, 1980), (DAUGMAN et al., 1985) como modelo de células simples do cortex visual e se baseia na descoberta da organização cristalina da principais células do cortex nos cérebros dos mamíferos (HUBEL; WIESEL, 1977). A função 2D de Gabor proposta por Daugman é um filtro de banda passante espacial que permite atingir o limite teórico para uma resolução associada às informações nos domínios 2D espacial e 2D de Fourier.

Gabor (GABOR, 1946) mostrou que existe um "princípio quantum" para informação: a associação do domínio tempo-freqüência para sinais 1D deve necessariamente ser quantificada de modo que nenhum sinal ou filtro possa ocupar menos do que uma certa área mínima. Essa área mínima, que reflete a inevitável troca entre a resolução do tempo e a de freqüência, tem um limite inferior de seu produto, análogo ao princípio de Heisenberg da incerteza na física. Ele descobriu que exponenciais complexas moduladas por uma Gaussiana proporcionam um resultado melhor.

A Equação 2 apresenta a forma geral do filtro 2D de Gabor usado em (ZHANG et al., 2003) para a extração de características da impressão da palma da mão.

$$G(x, y, \theta, u, \sigma) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{x^2+y^2}{2\sigma^2}} \exp^{2\pi i(uxcos\theta+uysen\theta)}, \quad (2)$$

Onde $i = \sqrt{-1}$, u é a frequência da onda senoidal, θ controla a orientação da função e σ representa o desvio padrão da função Gaussiana.



(a) *Parte real* (b) *Parte imaginária*

Figura 22: Resposta ao impulso do filtro 2D de Gabor

A Figura 22a mostra a parte real do filtro 2D de Gabor e Figura 22b mostra a parte imaginária. Para tornar o filtro mais robusto contra a luminosidade, o mesmo é transformado em zero CD (corrente direta) aplicando a Equação 3.

$$G'[s, y, \theta, u, \sigma] = G[x, y, \theta, u, \sigma] - \frac{\sum_{i=-n}^{n} \sum_{j=-n}^{n} G[i, j, \theta, u, \sigma]}{(2n+1)^2}, \quad (3)$$

Onde $(2n+1)^2$ é o tamanho do filtro. Por conta da simetria, a parte imaginária do filtro já possui zero DC. Certamente, o sucesso da extração do código depende da escolha dos parâmetros θ, u e σ. Em (ZHANG et al., 2003) aplicou-se um processo de refinamento para otimizar esses parâmetros e encontrou-se como melhor configuração: $\theta = \pi/4$, $u = 0,0916$ e $\sigma = 5,6179$. Esses valores foram usados na implementação do filtro 2D de Gabor utilizado neste projeto.

4.1.2.2 *Código binário da palma da mão*

Após a aplicação do filtro 2D de Gabor, o resultado é uma matriz de números complexos. O tamanho da matriz do *PalmCode* é definido como 32×32. O último passo da codificação leva em conta apenas a fase de cada um dos números complexos. São gerados dois códigos, um código para a parte real resultante e outro para a parte imaginária, observando apenas em qual quadrante o número complexo se mapeia. Se a parte real do número imaginário for maior do que zero, o código receberá o valor 1, e 0 caso contrário. A mesma lógica é

aplicada para a parte imaginária. O resultado final é definido por duas matrizes de 32×32 *bits*.

A Figura 23 mostra dois exemplos de *PalmCode*. As Figuras 23a e 23d mostram a área de interesse (como descrito na Seção 4.1.1) de duas imagens capturadas de uma mesma palma. A segunda coluna mostra a parte real e a terceira coluna a parte imaginária extraídas de suas respectivas imagens. Após serem extraídos, os códigos são comparados.

(a) *Palma A* (b) *Parte real A* (c) *Parte imaginária A*

(d) *Palma B* (e) *Parte real B* (f) *Parte imaginária B*

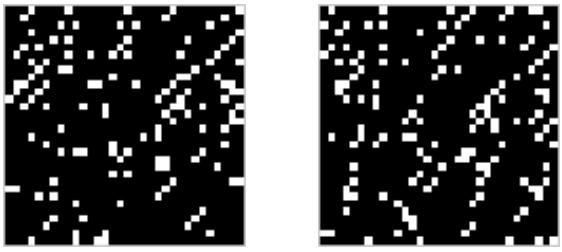
Figura 23: Exemplos de extração de *PalmCode*

4.2 Comparação

A comparação entre dois *PalmCodes* é efetuada utilizando o conceito de distância de Hamming (DH). Trata-se de um algoritmo simples que calcula a diferença entre dois códigos binários, aplicando a função XOR (ou exclusivo) entre as partes dos códigos. A Equação 4 define a distância de Hamming normalizada, onde $\oplus$ é o operador binário XOR, P e Q são os dois *PalmCodes* a serem comparados e N^2 o tamanho da parte real ou imaginária do *Palmcode*. Trata-se de uma porcentagem de *bits* diferentes entre dois *PalmCodes*.

$$DH = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} P_R(i,j) \oplus Q_R(i,j) + P_I(i,j) \oplus Q_I(i,j)}{2N^2} \tag{4}$$

Considere os códigos binários das palmas A e B da Figura 23. A Figura 24 mostra o resultado da aplicação da operação XOR entre eles, onde os *pixels* em branco representam os pontos em que os códigos diferem.



(a) *Parte real* (b) *Parte imaginária*

Figura 24: Resultado da aplicação do operador XOR

Calculando a distância de Hamming normalizada, ou seja, contando os pontos em brancos e dividindo pelo total de pontos, encontramos uma distância de 12,99. Essa distância confirma o fato de que as duas imagens foram obtidas de uma mesma palma. Note que distância de Hamming pode variar de 0 à 100. Resultados mais próximos de 0 indicam grande semelhança entre amostras.

Muitas vezes, o resultado da comparação não é tão próximo a 0 quando trata-se de imagens de uma mesma palma da mão. Isso ocorre, principalmente, devido ao diferente posicionamento da mão quando é feita a aquisição da imagem. Mesmo uma pequena diferença pode atrapalhar o processo de identificação, gerando resultados incorretos. Para não ocorrer esse problema, pode-se fazer outras verificações ao calcular a distância de Hamming variando a posição relativa entre as duas matrizes binárias (*PalmCode*). Essas variações podem ser executadas realizando deslocamentos de *bits* do *PalmCode* tanto no eixo horizontal quanto no eixo vertical. A Tabela 4 mostra a variação do resultado da distância de Hamming utilizando os deslocamentos. Note que são realizados todos os deslocamentos horizontais possíveis (à esquerda, nenhum e à direita) combinados com os deslocamentos verticais (para baixo, nenhum e para cima), totalizando 9 casos possíveis. No restante deste capítulo, a verificação feita considerando os deslocamentos de 1 *bit* em todos os sentidos é chamada de comparação com translação de 1 *bit*.

No exemplo proposto, o melhor resultado (19,50), ou seja, a menor distância ocorreu utilizando o deslocamento horizontal de 1 *bit* à esquerda e nenhum deslocamento vertical contra 33,44 no caso em que não foi aplicado nenhum deslocamento. Vale ressaltar

Tabela 4: Comparação com translação de 1 *bit*

Deslocamento				DH (%)
Horizontal		Vertical		
à esquerda	−1	para baixo	−1	28,40
à esquerda	−1	nenhum	0	**19,50**
à esquerda	−1	para cima	+1	32,88
nenhum	0	para baixo	−1	28,07
nenhum	0	nenhum	0	33,44
nenhum	0	para cima	+1	44,70
à direita	+1	para baixo	−1	40,06
à direita	+1	nenhum	0	45,26
à direita	+1	para cima	+1	50,36

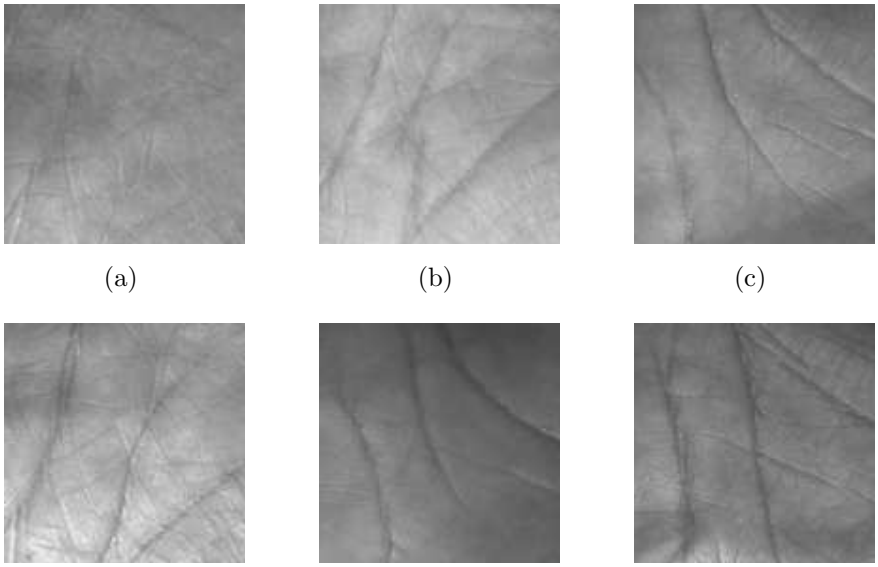
que os deslocamentos horizontais de 1 *bit* à esquerda, nenhum e à direita são relacionados como deslocamento horizontal de -1, 0 e +1 *bit*, respectivamente, assim como os deslocamentos verticais de 1 *bit* para baixo, nenhum e para cima são chamados de deslocamentos verticais de -1, 0, +1 *bit*. De forma análoga, pode ser feita a comparação com translação de 2 *bits* que irá considerar todas as combinações possíveis dos deslocamentos de -2, -1, 0, +1 e +2 nos eixos horizontal e vertical, totalizando 25 casos possíveis.

4.3 Banco de dados utilizado

As imagens utilizadas no projeto são provenientes do banco de dados da Universidade Politécnica de Hong Kong e está disponível em (POLYU, 2008). O banco de dados contém 8000 amostras de imagens de impressão da palma de 400 diferentes mãos.

As amostras foram feitas em duas seções de aquisição, sendo 10 amostras adquiridas na primeira seção e mais 10 na segunda. O tempo médio entre duas seções foi de um mês para que fosse possível ocorrer mudanças. O processo de delimitação e corte da área de interesse apresentado na Seção 4.1.1 foi realizado pela própria universidade, sendo disponibilizadas apenas as imagens finais da área de interesse de cada amostra.

A Figura 25 mostra uma série de exemplos de amostras de impressões da palma da mão de diferentes mãos contidas no banco de dados. Todas as imagens são em escala de cinza e possuem dimensão 128×128 *pixels*. Apesar de o banco de dados possuir tanto imagens de aquisição 2D quanto 3D, neste trabalho foram utilizadas apenas as imagens da aquisição 2D pois o algoritmo implementado não considera os dados da aquisição 3D.



(a) (b) (c)

(d) (e) (f)

Figura 25: Amostras contidas no banco de dados POLYU

4.4 Implementação

A proposta do projeto é usar o SmartCard para fazer a verificação e com isso aumentar a segurança. Para o caso da biometria da palma da mão, levando em consideração o método escolhido, a comparação é feita utilizando a distância de Hamming. O *PalmCode* deve ser passado para o cartão para que seja feito o armazenamento da biometria do indivíduo que possui o cartão e outro *PalmCode* deverá ser transferido para o cartão para que seja feita a confirmação da identidade do indivíduo através do cálculo da distância de Hamming.

O *PalmCode* é representado por 2 matrizes 32×32 *bits*, sendo uma para a parte real e outra para a parte imaginária. Logo, a tamanho total é de $2 \times 32 \times 32 = 2048$ *bits* ou 256 *bytes*. Por conta do limite de transmissão do JavaCard de 128 *bytes*, a transferência do código deve ser feita em duas etapas. Vale ressaltar que não há problemas para armazenar o código visto que os Smart Cards atuais possuem mais de 100kb de EEPROM e de 1 à 3Kb de RAM.

O JavaCard não permite o uso de matrizes. Portanto, o *PalmCode* é alocado em forma de vetor para a posterior comparação. Os *bytes* são comparados com a operação XOR utilizando iterações para percorrer o vetor onde estão alocados. O número de *bits* com valor 1 são contados e o resultado da soma é dividido pelo número total de *bits*. Essa divisão resulta em um número decimal mas não existem tipos de variáveis em Java

Card que permitem essa precisão. O tipo mais complexo de variável é o *short* (2 bytes). Considere a Equação 4, afim de evitar o problema da precisão, o dividendo é multiplicado por 100 antes que ocorra a divisão final, resultando em uma DH com precisão de duas casas. Note que o valor máximo do dividendo é de 2048 e 204800 após a multiplicação por 100. Isso leva à um problema de *overflow* pois as variáveis do tipo *short* podem variar apenas de -32768 à 32767. Para resolver o novo problema, o dividendo é multiplicado por 10 enquanto o divisor é divisor é dividido por 10 antes que a divisão final ocorra. Dessa forma, o valor máximo do dividendo será de $2048 \times 10 = 20480$ e o valor do divisor será de $2048/10 = 204$, evitando qualquer tipo de problema. para então ocorrer a divisão que dará a distância de Hamming em porcentagem.

O pseudocódigo do algoritmo que calcula a distância de Hamming com as modificações propostas é apresentado no Algoritmo 4, onde N^2 é o tamanho da matriz que foi armazenada como um vetor do tipo *short* com 64 posições, T representa o código binário armazenado no cartão e E representa o código binário do indivíduo que está requisitando a autenticação. Como visto na Seção 4.1.2, o *PalmCode* possui uma parte real e uma imaginária que são representados como (T_R, T_I) e (E_R, E_I), respectivamente.

Algoritmo 4 Algoritmo da Distância de Hamming entre *PalmCodes*

entrada E
saída $resultado$
1: $Nbits = 0$
2: **para** $i := 1 \rightarrow N^2$ **faça**
3: $\quad xored = T_R(i) \oplus E_R(i)$
4: $\quad Nbits = Nbits + contarBits(xored)$
5: $\quad xored = T_I(i) \oplus E_I(i)$
6: $\quad Nbits = Nbits + contarBits(xored)$
7: **fim para**;
8: $Nbits = 10 \times Nbits$
9: $resultado = Nbits/204$

A contagem dos *bits* feita no Algoritmo 4, nas linhas 6 e 8, pode ser implementada de diversas maneiras. A maneira mais simples e intuitiva de se fazer essa contagem é usando um laço somando *bit* a *bit* caso sejam iguais a 1, sempre fazendo 16 iterações. Para uma comparação mais rápida foi usado o Algoritmo 5, que possui o número de iterações igual ao número de *bits* iguais a 1.

Dessa forma, nem sempre será necessário fazer 16 iterações para retornar o resultado. No caso de uma variável sem 1s, não ocorre nenhuma iteração e o resultado é

Algoritmo 5 Algoritmo para contagem eficiente de *bits* em 1

entrada $xored$
saída $conteBits$
1: $conteBits = 0$
2: **enquanto** $xored! = 0$ **faça**
3: $xored = xoredAND(xored - 1)$
4: $conteBits = conteBits + 1$
5: **fim enquanto**;

retornado como *zero*. Utilizando esse código houve uma diminuição no tempo de execução da comparação.

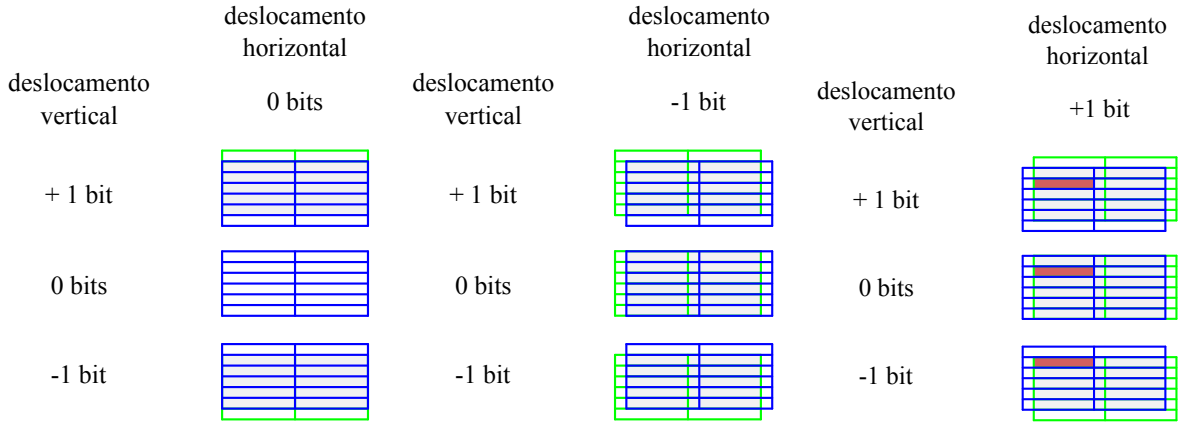
Como explicado na Seção 4.2, é possível realizar deslocamentos entre os *PalmCodes* comparados para melhorar o resultado da comparação. A implementação da comparação com translação de *bits* levou em conta a forma de alocação em memória, ou seja, em forma de vetores ao invés de matrizes. A translação foi feita admitindo que a cada 32 *bits* se iniciava uma nova linha. Outro ponto a ressaltar é o número de *bits* comparados. Sem translações, o número de *bits* comparados é igual a $32 \times 32 \times 2 = 2048$. Quando o deslocamento ocorre, o número de *bits* comparados diminui. Como exemplo, aplicando o deslocamento de 1 na horizontal (deslocamento à direita) e 0 na vertical (nenhum deslocamento), o número de *bits* comparados será $31 \times 32 \times 2 = 1984$.

Note que após os deslocamentos na direção vertical, a comparação dos *PalmCodes* resultantes continua baseada em comparações de *short* com *short*. No entanto, os deslocamentos da direção horizontal fazem uso da operação de deslocamento de *bits* ($\ll$ e $\gg$), acarretando comparação de *bits* que pertencem à diferentes shorts.

A translação de 1 *bit* consiste em realizar o deslocamento de -1 à 1 *bit* nas direções horizontal e vertical, resultando em 9 comparações entre *PalmCodes*. Para melhorar o desempenho da comparação com translação de *bits*, são consideradas todas as translações na vertical para cada translação na horizontal. Esse procedimento evita a execução de deslocamentos de *bits* de uma mesma linha do código binário diversas vezes. Apesar de exigir três vezes a memória utilizada para alocar as variáveis auxiliares, que acumulam resultados temporários, não chega a ser significativa. Para translação de mais *bits*, é necessário usar mais contadores temporários.

A Figura 26 ilustra o processo de translações em um *PalmCode* ilustrando somente 6×32 *bits*. Note que um *PalmCode* completo possui 32 linhas. Considere o conjunto de retângulos verdes como sendo o *PalmCode* de entrada e o conjunto de retângulos azuis

sendo o armazenado. Cada retângulo interno representa uma entrada de um vetor do tipo *short* (de 16 *bits*), totalizando 32 *bits* por linha. Na primeira iteração (Figura 26a), nenhuma translação é feita na horizontal. Logo, nenhuma operação de deslocamento de *bits* é necessária. A segunda iteração (Figura 26b) usa a operação de deslocamento de *bits* à esquerda e a terceira iteração à direita.



(a) *Primeira iteração* (b) *Segunda iteração* (c) *Segunda iteração*

Figura 26: Distância de Hamming com translações

A Figura (Figura 26c) possui um realce em vermelho representando uma das entradas (*short*) do vetor armazenado após a aplicação da operação de deslocamento à direita do vetor de entrada. A extração dos *bits* que formam a célula é realizada apenas 1 vez. Na mesma iteração, são feitas as comparações para os três casos na vertical.

4.5 Resultados

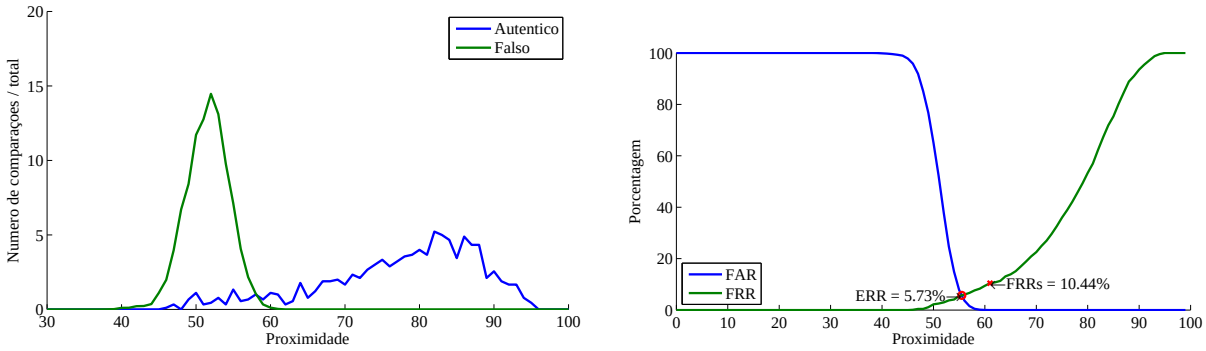
Para facilitar a comparação dos resultados obtidos nos testes da biometria da palma da mão com os de outras biometrias, é usado no restante deste capítulo a *proximidade* definida como 100 − Distância de Hamming, que indica em porcentagem a similaridade entre dois *PalmCodes*. Para testar e validar a implementação do código de comparação no cartão são usadas 10 amostras de 20 diferentes mãos selecionadas de forma aleatória do banco de dados introduzido na Seção 4.3.

Tendo em vista as 10 amostras de 20 diferentes mãos, são consideradas no total 200 amostras. Logo, 200×200 comparações são possíveis. As comparações usadas durante os testes são referenciadas como autênticas ou falsas, as comparações entre diferentes amostras de uma mesma mão são chamadas de autênticas enquanto as amostras de mãos

diferentes são chamadas de falsas. Para cada mão, são possíveis 10×10 comparações autênticas. Considerando todas as 40000 comparações possíveis, existem 2000 comparações autênticas e 38000 comparações falsas. As 200 comparações feitas entre códigos iguais, i.e., mesma amostra da mesma mão, são consideradas apenas para a avaliação do tempo de execução e são desconsideradas quando analisados os resultados das comparações, visto que sempre são autenticadas com uma proximidade de 100%. Nas próximas seções, os resultados obtidos com e sem translações de *bits* são apresentados e analisados.

4.5.1 Comparação direta

O primeiro teste foi feito comparando os *PalmCodes* sem qualquer tipo de deslocamento. A Figura 27a mostra a distribuição dos resultados das comparações autênticas e falsas realizadas. Para que as distribuições das comparações autênticas e falsas possam ser apresentadas em um mesmo gráfico, a distribuição foi ilustrada em referência ao total de comparações respectivas.



(a) *Distribuição das Proximidades* (b) *FAR e FRR para diferentes limites*

Figura 27: Resultados das comparações sem translação

O gráfico mostra que as comparações falsas se concentram de forma harmônica entre as proximidade de 40 e 60, enquanto as comparações autênticas ficaram distribuídas entre 45 e 95 com maior concentração entre 80 e 90. Como abordado na Seção 3.5, o resultado ideal ocorre quando as duas curvas não apresentam proximidades em comum. Esse gráfico pode auxiliar na escolha de um limite que permite identificar os resultados que devem ser considerados como originados de uma comparação da mesma mão.

A Figura 27b mostra o resultado dos testes de uma forma mais elaborada. Para cada escolha de proximidade, o gráfico mostra qual o FRR e FAR (conforme definições introduzidas na Seção 3.5). A definição do limite de proximidade mais adequado que

separa as comparações aprovadas das reprovadas também pode ser feita a partir dos dados desse gráfico. Para melhorar a segurança de um sistema é imprescindível que diferentes indivíduos não sejam confundidos. Logo, FAR deve ser o menor possível, mas sem grande comprometimento de FRR. No gráfico são destacados dois pontos importantes: o ponto em que as porcentagens são iguais (ERR) e o ponto em que FAR é menor que 0,1% (FRR seguro).

O gráfico mostra que o ERR se aproxima de 5% e FRR seguro fica próximo a 10%. O segundo ponto é o mais interessante para um sistema de segurança e significa dizer que, no caso da escolha da proximidade de 68% como limite do acesso, o resultado seria que a cada 1000 comparações de diferentes mãos, apenas uma seria considerada autêntica enquanto um indivíduo teria 90% de chance de conseguir seu acesso correto em cada tentativa.

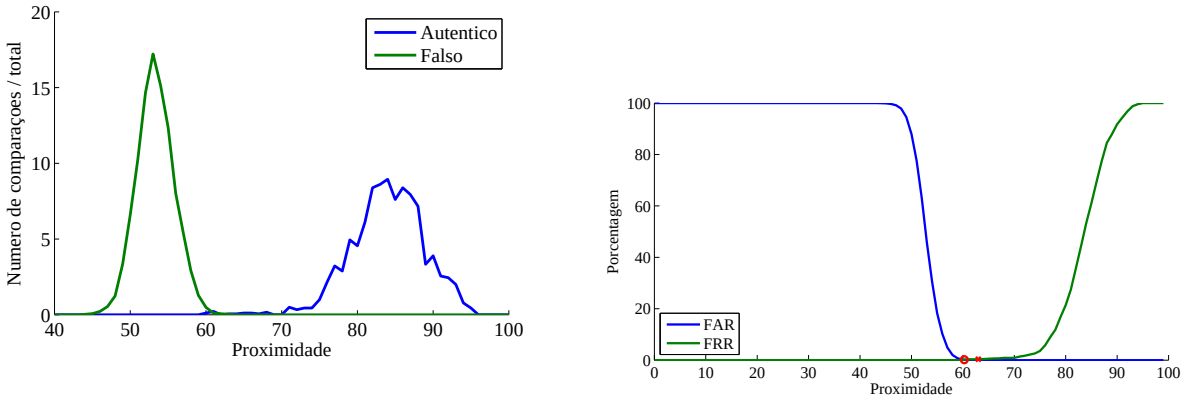
Conforme abordado na Seção 4.2, a proximidades pode ser diminuída caso sejam realizados deslocamentos de *bits* em um dos *PalmCodes* antes da comparação. Os resultados obtidos utilizando a translação de 1 e 2 *bits* serão mostrados na Seção 4.5.2.

4.5.2 Comparação usando translações

Para o mesmo conjunto de amostras usado no teste com comparações sem translação, foi realizado o teste utilizando comparações com translação de 1 *bit*. Conforme a explicação introduzida na Seção 4.2, a translação de 1 *bit* significa que um dos *PalmCodes* a serem comparados é deslocado 1 *bit* em todas as direções antes de cada comparação. Foram testadas todas as combinações deslocando -1, 0 e 1 nas direções vertical e horizontal. Ao todo são 9 combinações (a Tabela 4 exibe um exemplo que lista essas combinações).

A Figura 28a mostra a distribuição dos resultados obtidos fazendo as comparações usando a translação de 1 *bit*. O gráfico mostra que as comparações falsas tiveram uma distribuição harmônica concentrada entre 45 e 60 e as comparações autênticas ficaram concentradas entre 70 e 95 com alguns valores entre 60 e 70. Quando comparado ao gráfico das comparações sem translação (Figura 27a), pode ser visto a grande melhoria dos resultados das proximidades tornando mais fácil a escolha de um limite de aceitação.

A Figura 28b mostra o gráfico de FAR e FRR dos resultados usando translação de 1 *bit*. A porcentagem de erros iguais caiu para 0,17% (comparado com o ERR=5,73% da Figura 27b) e o FRR seguro caiu para 0,39% (comparado com o ERR=5,73% da Figura

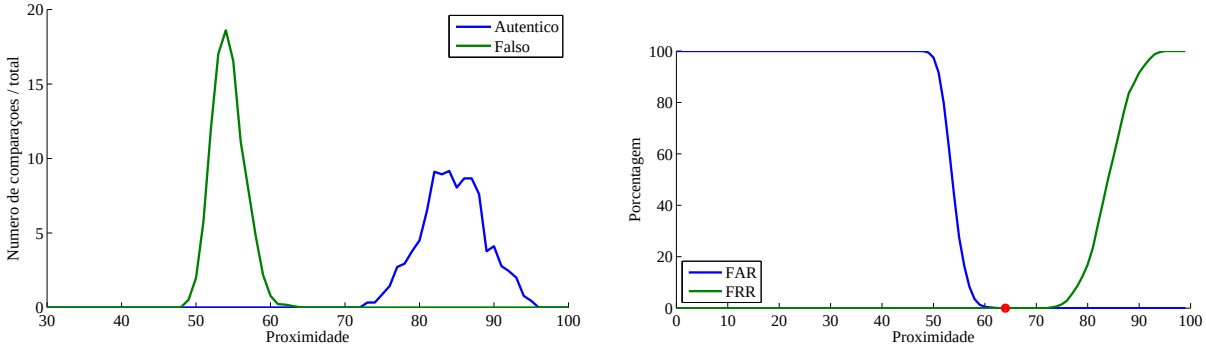


(a) *Distribuição das Proximidades* (b) *FAR e FRR para diferentes limites*

Figura 28: Resultados das comparações com translação de 1 *bit*

27b). Validando assim a possibilidade de fazer uso da translação de *bits* para melhorar o resultado da comparação.

Para obter uma melhoria ainda maior, é possível fazer a translação de mais *bits*. A Figura 29a mostra o gráfico da distribuição de proximidades de todas as comparações com translação de 2 *bits*, ou seja, variando o deslocamento em ambas as direções vertical e horizontal de -2 à 2 *bits*, totalizando 25 combinações, i.e., 16 comparações além das 9 também realizadas em comparações com translação de 1 *bit*.



(a) *Distribuição das Proximidades* (b) *FAR e FRR para diferentes limites*

Figura 29: Resultados das comparações com translação de 2 *bits*

O gráfico mostra que a distribuição é ideal, ou seja, os resultados das comparações autênticas e falsas estão totalmente separadas, o que torna ideal para a escolha de um limite de aceitação. A distribuição dos resultados da comparações autênticas está concentrada entre 75 e 95. Um fato interessante, mas esperado, a ser observado é o estrei-

tamento da distribuição de comparações falsas. Com várias translações, as comparações falsas tendem a alcançar um menor resultado de proximidade. Enquanto sem translação, esta distribuição se concentrava entre 40 e 60, com translação de 1 *bit*, a distribuição se concentra entre 45 e 60 e com translação de 2 *bits* se concentra entre 50 e 60. Para cada tentativa existe a chance de melhorar o resultado da comparação mas comparações falsas não alcançam proximidades muito maiores que 60%.

A Figura 29b mostra o gráfico que relaciona FAR e FRR para diferentes proximidades. Dessa vez, ambos os pontos importantes foram iguais a zero. Isso indica que a biometria se aproxima da perfeição. Ou seja, não haveria erros na comparação de diferentes indivíduos e também não haveria falhas nas tentativas autênticas.

Apesar do estudo dos resultados ser importante para validar o desempenho do algoritmo, não pode ser o único considerado. O tempo de execução é um outro aspecto importante pois irá indicar o tempo de espera do indivíduo a cada tentativa de autenticação.

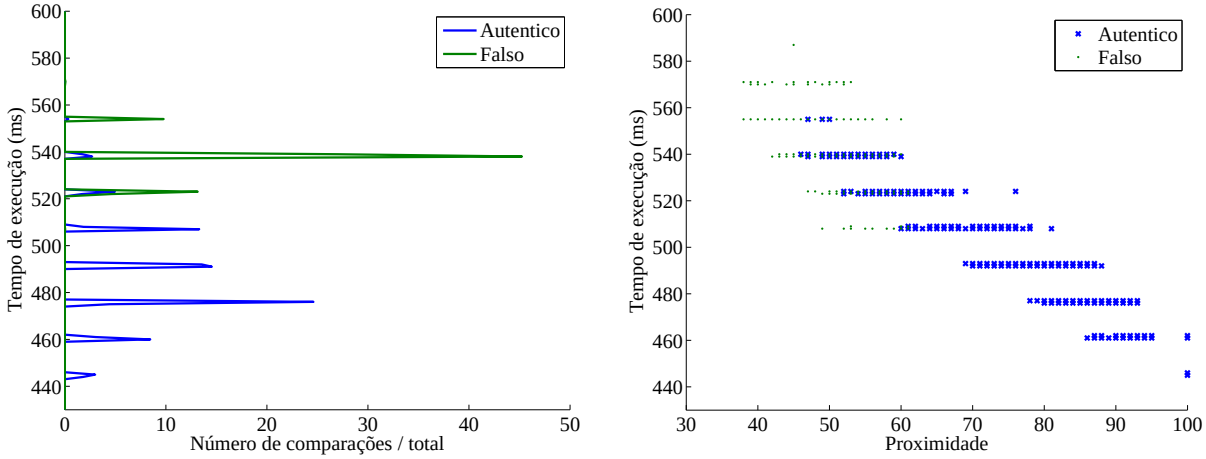
4.5.3 Tempo de execução

O tempo de transmissão e armazenamento em EEPROM de cada parte do *PalmCode* é de 850ms em média. Como explicado no início da Seção 4.2, devem ser feitas duas transmissões para armazenar o código. Logo, o tempo total de armazenamento é de 1700ms em média. Esse tempo é relativamente alto quando comparado ao tempo de transmissão e armazenamento em RAM de parte do *PalmCode* a ser comparado (185ms). Apesar de ser relativamente demorado, o tempo de armazenamento em EEPROM não é significativo pois trata-se de um processo que será realizado somente uma única vez durante o ciclo de vida do cartão.

O tempo total de comparação leva em conta a transmissão completa do *PalmCode* requerente e o processamento da distância entre ele e o *PalmCode* armazenado. Comparações com translação de *bits* podem ser feita para atingir ótimas taxas de erro mas certamente impacta no tempo de execução uma vez que a comparação com translação de 1 *bit* considera 9 combinações de deslocamentos e 25 noas comparações com translação de 2 *bits*. Lembrando que para cada combinação deve ser calculada a distância de Hamming.

A Figura 30 mostra o tempo de execução das comparações sem translação de *bits*. A Figura 30a mostra a distribuição do tempo de execução para todas as 40000 comparações

(sendo 2000 autênticas e 38000 falsas) realizadas. Novamente, os valores normalizados (a porcentagem) foram usados para que fosse possível ilustrar todos os tempos em um único gráfico. A Figura 30b mostra a relação entre o tempo de execução e o resultado da comparação. O tempo de execução das comparações autênticas variou entre 440 e 560 ms enquanto o tempo de execução das comparações falsas variou de 500 à 580 ms.



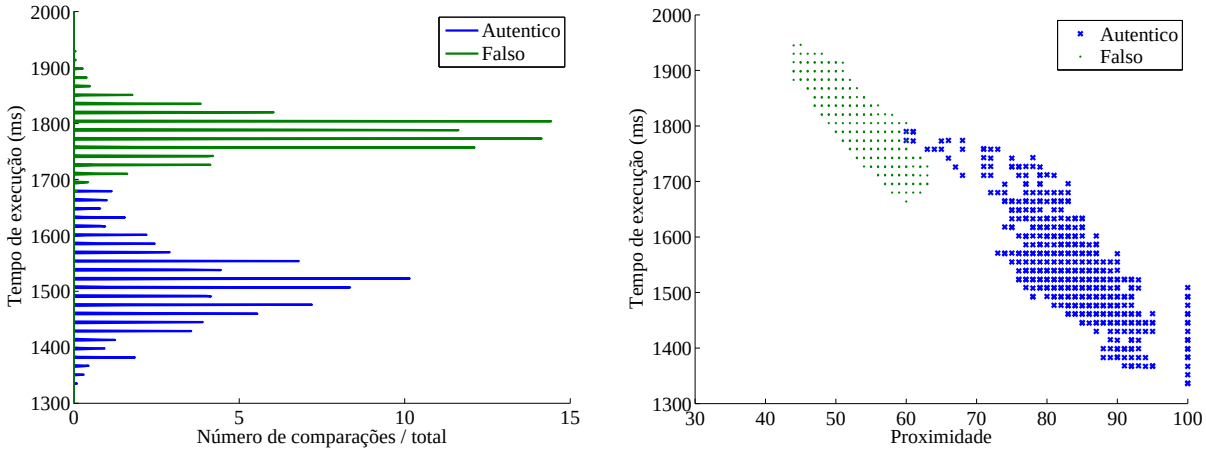
(a) *Tempo de execução* (b) *Proximidade vs. tempo de execução*

Figura 30: Tempo de execução para comparações sem translação

As mesmas relações mostradas para os tempos de execução das comparações sem translação de *bits* são mostradas para as comparações com translação de 1 *bit* nos gráficos da Figura 31. Novamente é mostrada a tendência dos tempos de execução das comparações autênticas (variando de 1350 à 1800 ms) serem menores que o das comparações falsas (variando de 1650 à 1950).

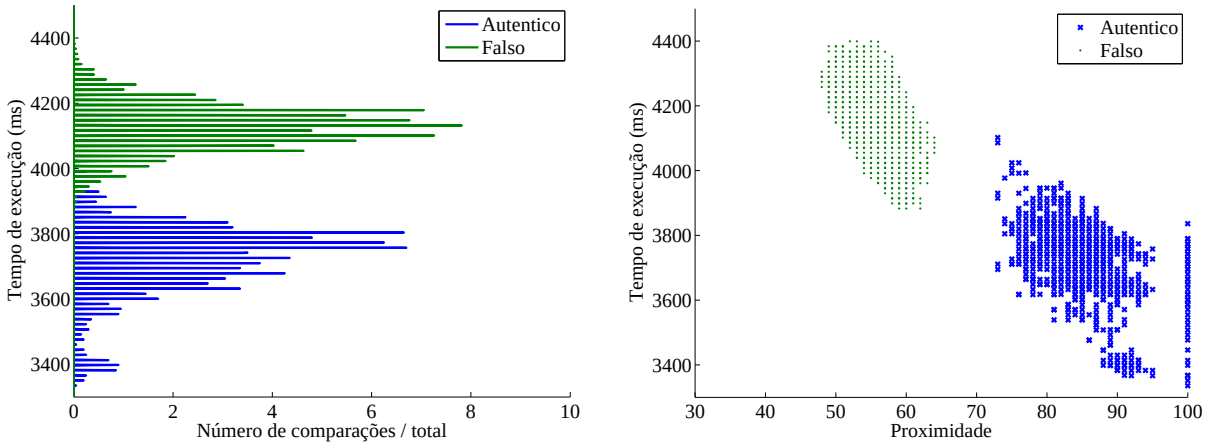
Os gráficos da Figura 32 mostram, de forma mais destacada, que os tempos de execução das comparações autênticas tendem a ser menor que os das comparações falsas. Note, na Figura 32a, que existem duas grandes concentrações para os tempos de execução das comparações autênticas e falsas.

Uma característica interessante a ser observada é a relação entre a diminuição do tempo de comparação e o aumento da proximidade. Essa relação se dá pelo algoritmo de contagem de *bits* usado (Algoritmo 5). Ele indica que, quanto mais semelhantes forem as entradas, menos iterações serão necessárias para a contagem dos *bits*, e isso resultará em um tempo de execução menor. Note que os pontos que possuem Proximidade de 100% são as amostras comparadas com elas mesmas. Essas comparações foram usadas apenas



(a) *Tempo de execução* (b) *Proximidade vs. tempo de execução*

Figura 31: Tempo de execução para comparações com translação de 1 *bit*



(a) *Tempo de execução* (b) *Proximidade vs. tempo de execução*

Figura 32: Tempo de execução para comparações com translação de 2 *bits*

na análise do tempo de execução e confirmam que as comparações mais semelhantes se completam em menor tempo de execução.

Tabela 5: Tempo de execução das comparações variando a translação

	Comparações autênticas		Comparações falsas	
Translação (*bits*)	Média (ms)	Desvio padrão	Média (ms)	Desvio padrão
0	489	22	538	8
1	1520	74	1783	37
2	3725	109	4127	79

A Tabela 5 mostra as médias e desvios padrão dos tempos de execução das comparações sem translação e com translação de 1 e 2 *bits*. A Figura 33 mostra o gráfico

comparativo entres os tempos de execução médios. O tempo de execução médio das comparações com translação de 2 *bits* é quase 7× maior do que o tempo de execução médio das comparações sem translação enquanto com translação de 1 *bit*, o tempo aumenta em aproximadamente 3×. Como sistemas de biometria usando cartão exigem uma resposta em tempo real, a translação de 2 *bits* apresenta um fator negativo por mais que faça com que os erros sejam reduzidos.

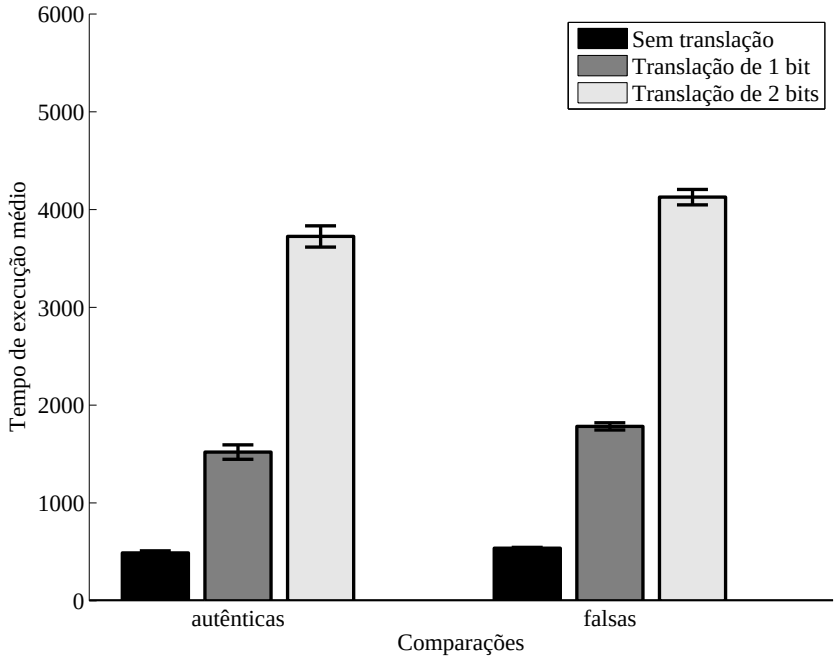


Figura 33: Tempo de execução das comparações variando a translação

Como visto anteriormente, a verificação sem translações requer apenas uma comparação entre os *PalmCodes*, enquanto a comparação com translação de 1 *bit* requer 9 comparações entre *PalmCodes* e com translação de 2 *bits* necessitam 25 comparações. Essa informação parece conflitar com os tempos de execução médios pois a relação entre o tempo de execução do algoritmo sem translação e com translação de 1 *bit* é apenas de 3× maior enquanto realiza 9× o mesmo trabalho computacional. Isso é explicado pela consideração do tempo de transferência (185 ms). A Tabela 6 mostra os tempos de execução médios desconsiderando o tempo de transferência, apresentando agora resultados mais coerentes em relação às translações. Como são feitas 2 transferências, 370 ms são apenas destinados à transferência.

A Figura 33 mostra o gráfico comparativo entres os tempos de execução médios desconsiderando o tempo de transferência do *PalmCode*. Note que agora os resultados

Tabela 6: Tempo de execução das comparações sem o tempo de transferência

	Comparações autênticas		Comparações falsas	
Translação (*bits*)	Média (ms)	Desvio padrão	Média (ms)	Desvio padrão
0	119	22	168	8
1	1150	74	1413	37
2	3355	109	3757	79

comparativos entre os tempos de execução médios tem relação mais semelhante ao esforço computacional necessário.

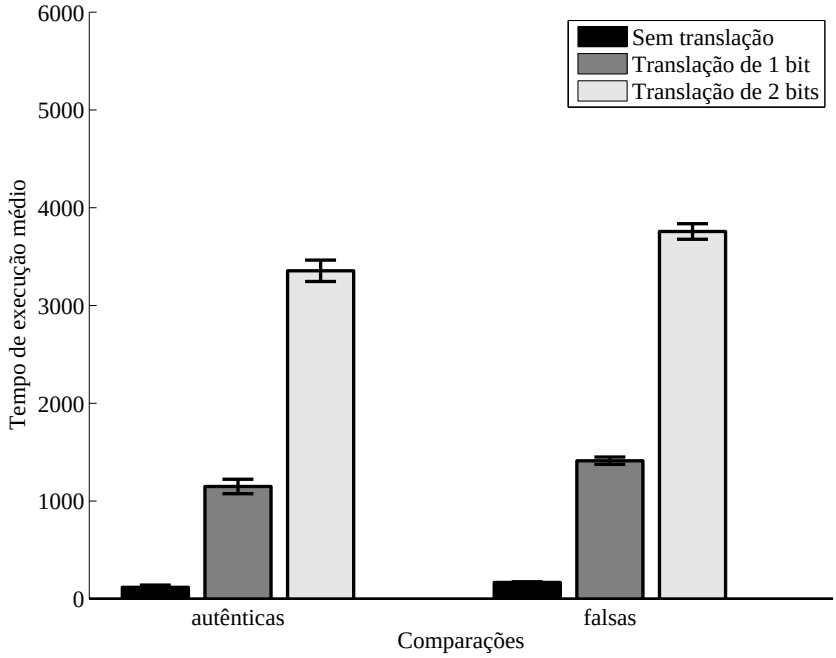


Figura 34: Tempo de execução das comparações desconsiderando o tempo de transferência do *PalmCode*

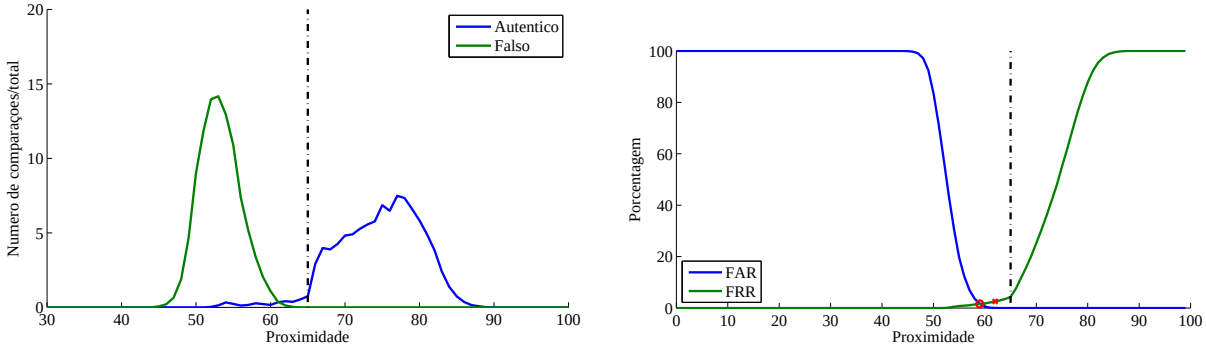
Os resultados apresentados indicam que um cartão é capaz de executar um método de biometria da impressão da palma da mão proporcionando alto desempenho e também uma confiabilidade alta, podendo ser estudado se o determinado sistema deverá priorizar a velocidade da comparação ou as baixas taxas de erro. A partir da análise dos resultados obtidos e afim de melhorar o tempo de execução, é possível usar as comparações com translação de *bits*, que obtiveram as menores taxas de erro, com um limite de aceitação definido para que o processamentos dos diversos deslocamentos de *bits* possa ser interrompido em caso de autenticação positiva e, com isso, diminuir o tempo de execução. A Seção 4.5.4 apresenta os resultados obtidos usando essa otimização.

4.5.4 Comparação com limite de aceitação

O limite de aceitação, nada mais é, que o valor da proximidade, a partir do qual, as comparações são consideradas corretas, i.e., após alcançada uma proximidade pré-definida, o cartão já terá a resposta da comparação, podendo encerrar o processamento e retornar a resposta. Definindo um limite de aceitação para abortar a execução do restante do código pode reduzir apenas o tempo de execução das comparações autênticas mas não afeta o tempo de execução das comparações falsas uma vez que, neste caso, a execução não será interrompida. As comparações sem translação de *bits* realizam o cálculo da DH apenas uma vez, logo, não podem ser interrompidas. O limite de aceitação foi implementado para as comparações com translação de 1 e 2 *bits*.

A escolha do limite foi baseada nos resultados das comparações com as translações de 0, 1 e 2. Os resultados das comparações com translação de 2 *bits* (Figura 29b) indicam que as comparações falsas não apresentam proximidades maiores que 65. Prezando pela segurança, esse valor de proximidade foi o limite de aceitação escolhido.

A inserção de um limite de aceitação não irá modificar os resultados dos testes, por isso, foram escolhidas novas amostras do banco de dados para realizar os testes. Foram usadas 20 amostras de 20 diferentes mãos. Totalizando 400 amostras e 160000 comparações, sendo 8000 comparações autênticas e 152000 comparações falsas. A Figura 35a mostra a distribuição dos resultados das comparações com translação de 1 *bit* e limite de aceitação. A reta indica a proximidade escolhida como limite. A Figura 35b mostra a relação entre diferentes Proximidades e taxas FAR e FRR.



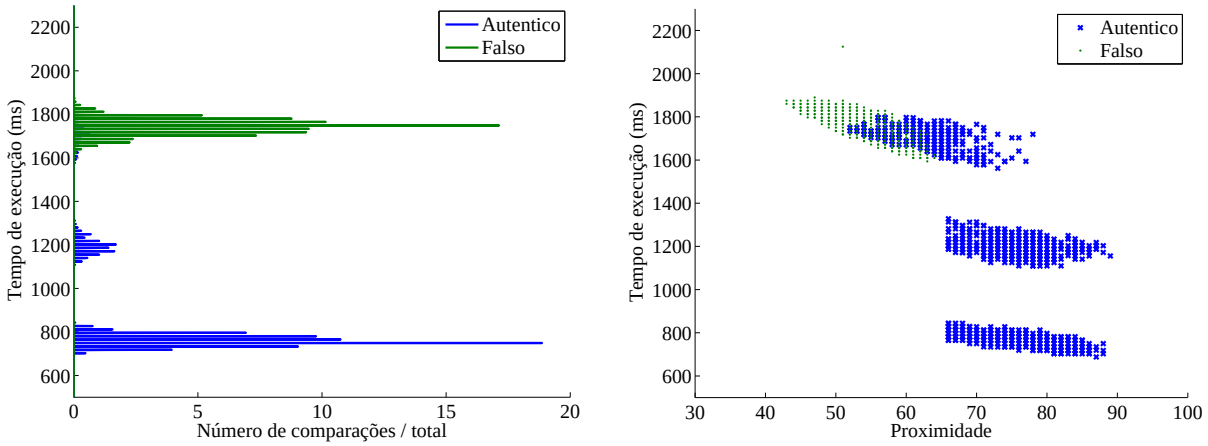
(a) *Distribuição dos resultados* (b) *FAR e FRR*

Figura 35: Resultados das comparações com limite de aceitação e translação de 1 *bit*

Para a proximidade escolhida como limite de aceitação, FAR foi de 0% e o FRR

igual a 3,55%. O aumento da taxa de erro em relação ao teste das comparações com translação de 1 *bit* se devem à escolha aleatória das novas amostras.É uma taxa pequena e segura em relação à comparações falsas. É possível notar que em ambos os gráficos ocorre um súbito aumento na inclinação da curva após o limite escolhido validando a existência do limite de aceitação uma vez que, a partir daquele ponto, não são obtidas melhorias no resultado. A Figura 35b indica que é possível diminuir o limite de proximidade sem grandes impactos no resultado da autenticação. O ponto em que FRR e FAR são iguais foi a porcentagem de 1,62% e o FRR seguro foi de 2.99%.

A Figura 36 apresenta os gráficos relacionados ao tempo de execução das comparações com translação de 1 *bit* e limite de aceitação. Nos gráficos ficam claros 3 agrupamentos dos tempos que indicam os valores aceitos e cada comparação. Como o número de comparações falsas são todas negadas, elas ficaram no grupo com maior tempo de execução enquanto as comparações autênticas aceitas se concentraram em dois grupos com menores tempos de execução. O tempo médio de execução das comparações autênticas foi de 868ms com desvio padrão de 249 enquanto o tempo de execução médio das comparações falsas foi de 1744ms com desvio padrão de 37, sendo o último similar ao tempo das comparações com translação de 1 *bit*, como esperado.

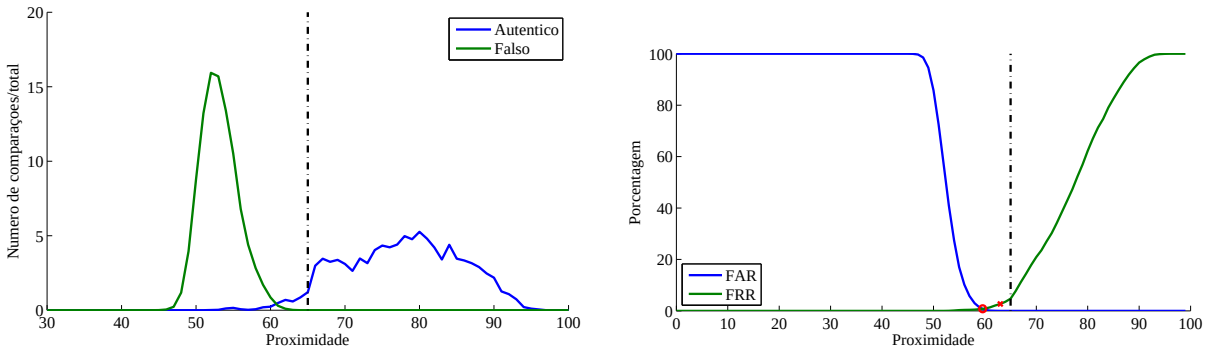


(a) *Tempo de execução* (b) *Proximidade em relação ao tempo*

Figura 36: Tempo de execução para comparações com limite de aceitação e translação de 1 *bit*

A Figura 37 apresenta os gráficos da distribuição da proximidade e do FAR e FRR em relação à proximidade para os testes das comparações com translação de 2 *bits* e limite de aceitação. Para a proximidade escolhida como limite de aceitação, FAR foi de 0% e

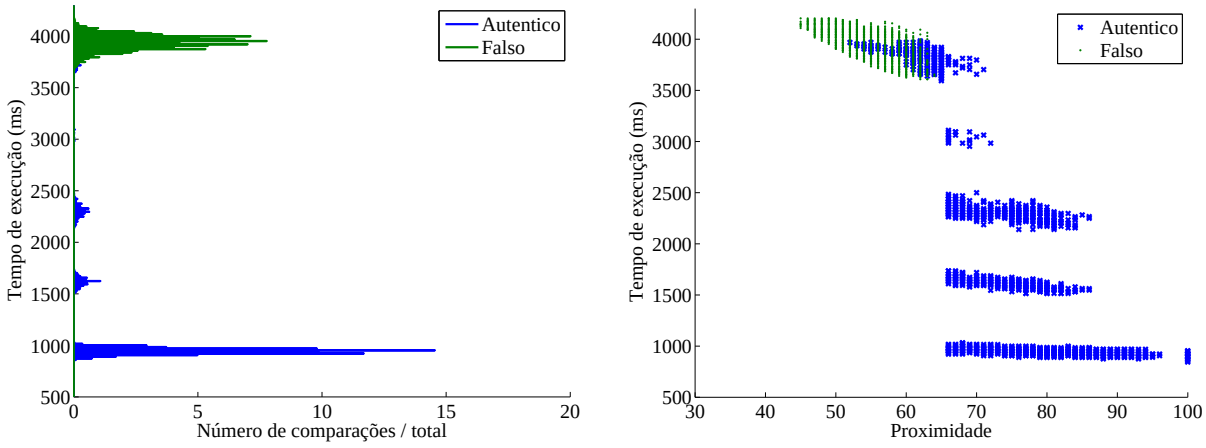
o FRR igual a 3,53%. Os valores foram iguais aos alcançados com as comparações com translação de 1 *bit* e limite de aceitação (FAR=0%, FRR=3,55%).



(a) *Distribuição dos resultados* (b) *FAR e FRR*

Figura 37: Resultados das comparações com limite de aceitação e translação de 2 *bits*

A Figura 38 apresenta os gráficos relacionados ao tempo de execução das comparações com translação de 2 *bits* e limite de aceitação. Existem 5 grupos que representam as 5 iterações que ocorrem. É possível notar que poucas comparações executaram as últimas duas iterações. O tempo de execução médio das comparações autênticas foi de 1217ms. Se comparado ao tempo de execução médio das comparações com translação de 1 *bit* e limite de aceitação, que foi de 868ms, pode-se concluir que o benefício é muito pequeno em relação ao custo de processamento.



(a) *Tempo de execução* (b) *Proximidade em relação ao tempo*

Figura 38: Tempo de execução para comparações com limite de aceitação e translação de 2 *bits*

Para comparações autênticas, o tempo de execução das comparações com transla-

ção de 1 e 2 *bit* tiveram uma redução considerável utilizando o limite de aceitação. No caso de uma implementação real em um cartão inteligente, é esperado que apenas o dono do cartão tente sua própria autenticação. Logo, o tempo de execução para as comparações autênticas possui uma relevância maior em relação às comparações falsas. Entretanto, as comparações falsas não podem possuir um tempo de execução muito alto porque isso tem um impacto negativo na aceitação da tecnologia. Levando isso em consideração, as comparações com translação de 1 *bit* e limite de aceitação obtiveram os melhores resultados.

4.6 Considerações Finais

O capítulo abordou todos os aspectos necessários para a implementação da biometria da impressão da palma da mão em um cartão inteligente. Foram vistos diferentes formas de executar a comparação entre dois códigos. Uma forma condicional usando limites de aceitação também foi implementada e analisada trazendo um balanceamento entre tempo de execução e confiabilidade do resultado da autenticação.

Capítulo 5

ÍRIS

A BIOMETRIA da íris vem ganhando significativa importância no mercado mundial. Esta biometria se destaca das demais por ser muito segura, de grande durabilidade. Como mencionado na Seção 2.3, John Daugman é considerado o pioneiro da biometria da íris pois apresentou o primeiro trabalho de grande aceitação na área. Seu trabalho é baseado no uso do filtro 2D de Gabor para extração das características da textura da íris e serviu de inspiração para diversos trabalhos subsequentes tanto na área de biometria da íris como em outras biometrias, por exemplo, o método biométrico da impressão da palma da mão implementado nesse projeto).

Devido a sua grande importância e robustez, o método de Daugman (DAUGMAN, 1993) foi escolhido para implementação da biometria da íris. O método de extração do código binário, o algoritmo de comparação e os bancos de dados utilizados são abordados nas Seções 5.1, 5.2 e 5.3, respectivamente. A Seção 5.4 trata a forma de implementação do algoritmo proposto e a Seção 5.5 apresenta e analisa os resultados obtidos nos testes.

5.1 Extração

A extração das características da íris é realizada em três principais passos: segmentação, normalização e formação do código binário (*IrisCode*) e os detalhes inerentes deste procedimento são abordados nas Seções 5.1.1, 5.1.2, e 5.1.3, respectivamente.

5.1.1 Segmentação

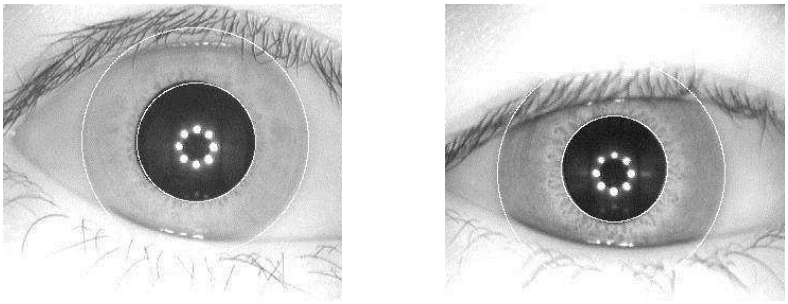
A Segmentação é provavelmente o passo mais importante e mais complicado da a extração pois qualquer erro invalidará todos os passos seguintes, inclusive a comparação para autenticação. Basicamente, o resultado esperado é a localização exata dos contor-

nos interno e externo da íris. A tarefa pode se mostrar extremamente desafiante quando os contornos estão muito encobertos pelas pálpebras ou pelos cílios. Outra dificuldade pode ser encontrada em íris de coloração muito claras pois podem ser confundidas com a esclerótica.

O algoritmo utilizado foi a transformada de Hough. Trata-se de um algoritmo padrão em processamento de imagens usado para determinar parâmetros de objetos geométricos simples. A transformada de Hough circular pode ser empregada para deduzir os pontos da coordenada do centro e o raio dos contornos da íris.

Antes da aplicação da transformada, é necessário identificar a extração das bordas utilizando algum filtro especial. No caso foi aplicada a derivada da primeira ordem da intensidade da imagem para encontrar todos os possíveis pontos de contorno que serão usados na Transformada de Hough. O ponto crucial nessa etapa é a escolha do limite do que será considerado como contorno.

A Figura 39 mostra exemplos de segmentações executadas com sucesso. Note que mesmo com o contorno muito encoberto é possível efetuar a segmentação correta da íris. Para a detecção das pálpebras é usada a Transformada de Hough linear.



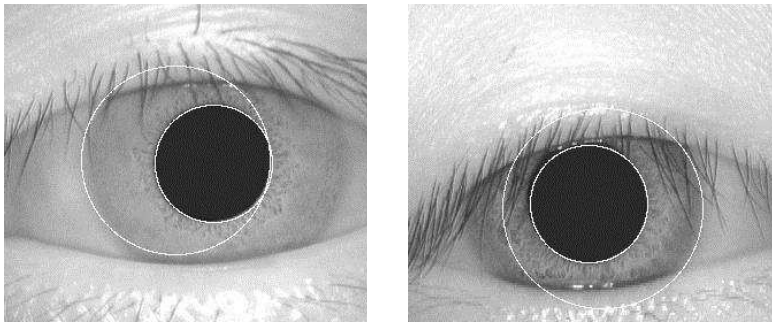
(a) *Menor dificuldade* (b) *Maior dificuldade*

Figura 39: Exemplos de segmentação sem falhas

É possível que ocorram erros na segmentação das imagens, ocasionando a geração de um *IrisCodes* de baixa qualidade e portanto prejudicando a comparação. A Figura 40 mostra 2 exemplos de falhas de segmentação.

5.1.2 Normalização

A forma circular da íris não favoriza a comparação. Daugman propôs a normalização da íris para torná-la retangular de dimensões fixas. A Figura 41 ilustra o processo de transformação



(a) (b)

Figura 40: Falhas de segmentação da íris

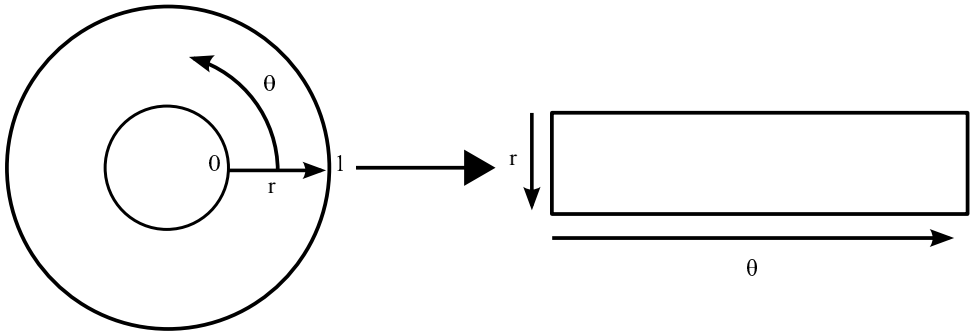


Figura 41: Normalização da íris

A normalização consiste em converter as coordenadas polares em coordenadas lineares, adequando o máximo e mínimo à um retângulo de tamanho fixo. Como o resultado é um retângulo de tamanho fixo, a diferença da espessura ocasionada pela dilatação e contração da íris é eliminada. O mesmo processo de normalização é efetuado também para a máscara que indica quais áreas são válidas como textura da íris.

5.1.3 Código binário da íris

O resultado da normalização ainda possui muitas informações e muita interferência da luminosidade, logo, não está ideal para comparação direta. Daugman então propôs a convolução da imagem utilizando o filtro 2D de Gabor (visto na Seção 4.1.2.1). O resultado da convolução é uma matriz de número complexos.

A ferramenta utiliza um método semelhante baseado no trabalho de Daugman. Porém, aplicou a convolução à imagem da íris normalizada utilizando o 1D Log-Gabor wavelets (MASEK et al., 2003). O resultado é uma matriz de números complexos com dimensão 8×128 bits.

A matriz é então codificada de acordo com a fase do numero complexo. O número complexo é substituído por 2 *bits* de acordo com sua localização. A Figura 42 mostra essa troca de forma visual. O *IrisCode* é composto de 2 partes: a parte código é uma matriz de números binários resultante desse processo e sua dimensão é 8×256 bits e a parte máscara é redimensionada para 8×256 também como matriz binária, onde 1 indica lugares em que a textura da íris é valida e 0 indica onde são localizados os obstáculos, como pálpebras e cílios.

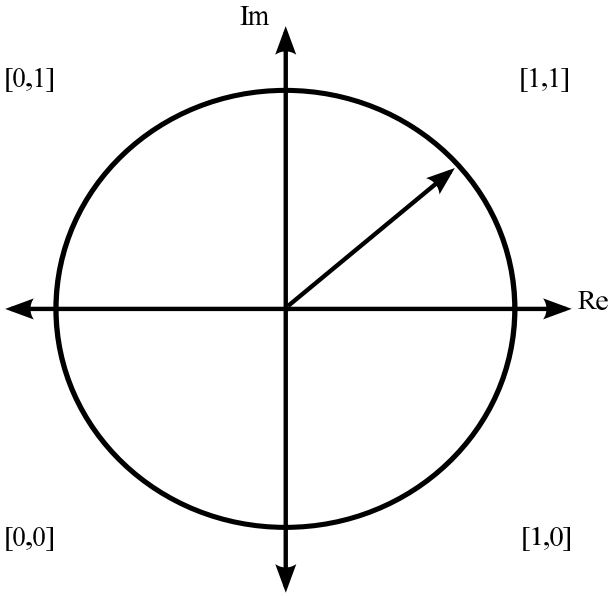


Figura 42: Codificação do *IrisCode*

5.2 Comparação

Conforme explicado, o *IrisCode* é composto por um código e uma máscara que indica os pontos onde seu código é válido. Ambos são representados por uma matriz de 8×256 *bits* totalizando 2048 *bits* cada. Para realizar a comparação entre dois *IrisCodes*, é utilizada a distância de Hamming entre estas. Um método semelhante foi visto na Seção 4.2 pois o algoritmo de comparação entre *PalmCodes* foi inspirado no método de Daugman para *IrisCode*.

A Distância de Hamming é o número de *bits* válidos diferentes entre os códigos comparados com relação ao total. A Equação 5 define a distância de Hamming para 2 *IrisCodes* A e B.

$$DH = \frac{||(C_A \oplus C_B) \cap M_A \cap M_B||}{||M_A \cap M_B||} \quad (5)$$

Onde, $\oplus$ é o operador binário XOR, $\cap$ o operador binário AND , A e B dois *IrisCodes* (compostos por código C e máscara M) a serem comparados.

A DH (Distância de Hamming) representa a distância entre dois *IrisCodes* em porcentagem mas apenas considera a comparação direta entre dois códigos. Para uma comparação mais segura, é possível considerar translações no código binário. Considere um código binário de um requerente autêntico a ser comparado com um código armazenado para tentativa de aceitação. A comparação direta pode falhar devido a uma pequena rotação da íris no momento da captura. A Tabela 7 ilustra o código em forma de matriz de 8×256 *bits*.

Tabela 7: *IrisCode* sem translação

	1	2	3	4	5	6	7	8	9	...	249	250	251	252	253	254	255	256
1	1	0	0	0	0	1	1	0	1	...	1	0	1	0	1	1	0	0
2	1	0	1	0	1	0	1	0	0	...	1	0	1	0	1	1	1	1
3	1	1	1	0	0	1	1	0	0	...	0	0	1	0	1	0	1	0
4	0	1	1	0	1	0	1	1	1	...	0	0	1	1	1	0	0	1
5	0	0	1	0	1	0	1	0	0	...	0	0	0	0	1	1	1	0
6	0	1	0	0	1	0	1	0	1	...	1	0	0	0	1	0	1	0
7	1	1	1	0	1	0	1	1	0	...	1	0	1	0	1	0	1	1
8	1	0	0	0	1	0	1	1	1	...	1	0	0	0	1	0	0	0

5.2.1 Translação de *bits*

Como visto na Seção 5.1.2, a imagem destacada da íris passa pelo processo de normalização no qual é transformada de círculo para retângulo. Logo, para termos o efeito de rotação da íris, é necessário deslocar as colunas entre as extremidades. A Tabela 8 mostra o mesmo *IrisCode* com deslocamento de -2 *bits* ou 2 *bits* à esquerda. O deslocamento deve sempre ser feito de 2 em dois pois um deles representa a fase real e o outro a fase imaginária do número complexo originalmente extraído da imagem.

Note que as duas primeiras colunas passaram a ser as últimas e com isso todas as colunas sofreram deslocamento de 2 *bits* para a esquerda. Vale lembrar que cada 2 *bits* representam a fase de um número complexo, como visto na Seção 5.1.3. O deslocamento de apenas 1 *bit* faria com que a parte real de um código fosse comparada com a parte imaginária de outro, o que certamente acarretaria em erros. Portanto, a translação de 1

Tabela 8: *IrisCode* com translação de 1 *bit* para esquerda

	1	2	3	4	5	6	7	8	9	...	249	250	251	252	253	254	255	256
1	0	0	0	1	1	0	1	1	1	...	1	0	1	1	0	0	**1**	**0**
2	1	0	1	0	1	0	0	1	0	...	1	0	1	1	1	1	**1**	**0**
3	1	0	0	1	1	0	0	0	1	...	1	0	1	0	1	0	**1**	**1**
4	1	0	1	0	1	1	1	0	0	...	1	1	1	0	0	1	**0**	**1**
5	1	0	1	0	1	0	0	0	0	...	0	0	1	1	1	0	**0**	**0**
6	0	0	1	0	1	0	1	0	1	...	0	0	1	0	1	0	**0**	**1**
7	1	0	1	0	1	1	0	1	0	...	1	0	1	0	1	1	**1**	**1**
8	0	0	1	0	1	1	1	0	0	...	0	0	1	0	0	0	**1**	**0**

bit à esquerda refere a ambas as partes real e imaginária e o código deverá ser deslocado em 2 *bits* à esquerda.

Esse novo código deslocado pode então ser comparado ao código armazenado usando a distância de Hamming na tentativa de obter melhores resultados. É importante ressaltar que a máscara deve passar pelo mesmo processo para que valide os *bits* corretos. A translação de n *bits* significa que foram testadas as translações de $-n$ à n *bits*, i.e., a translação de 2 *bits* indica que serão testadas todas as 5 translações possíveis de -2 à 2.

5.3 Bancos de Dados

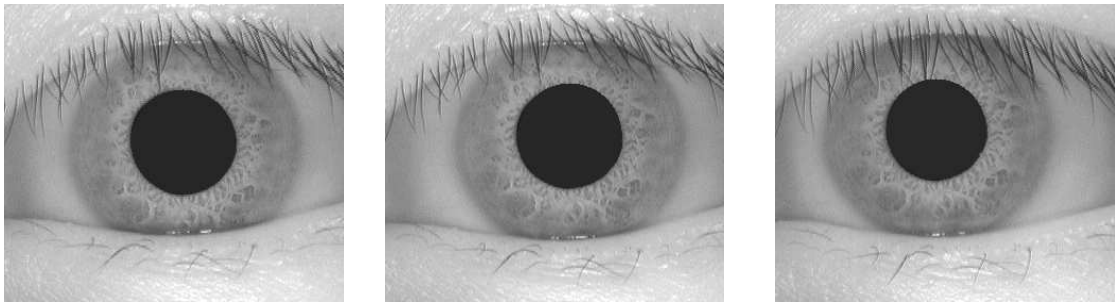
Para fazer os testes dos algoritmos de comparação implementados no cartão inteligente, foram usados dois diferentes bancos de dados. Eles possuem diferentes características em suas aquisições. Ambos os bancos foram coletados pela CASIA (*Chinese Academy of Sciences' Institute of Automation*) e estão disponíveis para *download* em seu site ((CASIA, 2004)).

5.3.1 CASIA Iris V1

O banco de dados em questão foi um dos primeiros disponíveis de forma gratuita para estudo da biometria da íris. Foi coletado usando uma câmera para captação de infravermelho. Possui 756 imagens provenientes de 108 diferentes olhos com 7 amostras cada. As imagens tem uma resolução de 320×280 e estão armazenadas no formato BMP.

De forma a proteger o projeto de captura, a parte interna da pupila, onde apareceriam as reflexões da luz, necessária para a captação das imagens, foi substituída por uma região escura de cor constante. Note que este processamento automático não afeta a íris.

Apesar de muito usado, o pós-processamento automático das imagens é apontado como um fator negativo para a avaliação de métodos de extração, uma vez que, torna a segmentação do círculo interno na íris mais simples (PHILLIPS; BOWYER; FLYNN, 2007).



(a) (b) (c)

Figura 43: Exemplos de imagens de íris do CASIA Iris V1

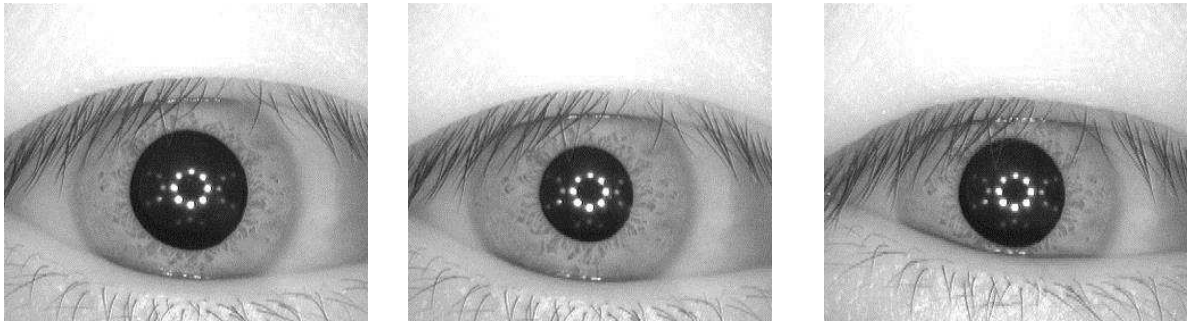
A Figura 43 mostra três imagens capturadas de um mesmo olho. Uma das características desse banco de dados é a pequena variação do olho durante a captura. Nas imagens dessa figura são mostradas as pequenas variações entre elas. A falta de desafio torna o banco de dados ideal para a validação do algoritmo de comparação mas se distancia do uso real da biometria.

5.3.2 CASIA Iris V4 Interval

Após a distribuição do CASIA Iris V1, a mesma instituição construiu novos bancos de dados com diversas características para diferentes tipos de estudos da biometria da íris. O CASIA Iris V4 Interval inclui 2369 imagens de 249 diferentes olhos capturados por uma câmera própria com aproximação. O número de repetições de cada olho é variável. As imagens tem resolução de 320×280 pixels e foram capturadas em duas seções.

A Figura 44 mostra três amostras do mesmo olho. As imagens foram escolhidas de modo a exemplificar a maior dificuldade proporcionada por este banco de dados. Isso faz com que a captura se torne mais similar ao uso real em que pequenas diferenças são esperadas.

Entre as outras opções distribuídas pela CASIA, foi escolhido o tipo Interval por possuir uma resolução menor, uma aproximação maior gerando assim imagens com alta nitidez da íris e por possuir imagens capturadas em 2 seções, que torna possível analisar a permanência da biometria.



(a) *Pupila mais dilatada* (b) *Pupila mais contraída* (c) *Íris mais encoberta*

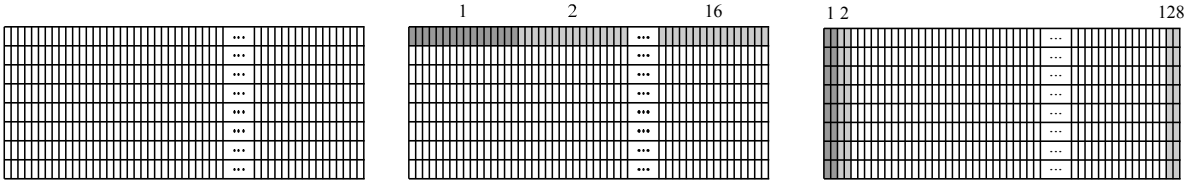
Figura 44: Exemplos de imagens de íris do CASIA Iris V4 Interval

5.4 Implementação

Para a implementação das comparações da biometria da íris no cartão inteligente foi usada a plataforma Java Card. A plataforma possui uma restrição na transferência dos dados para o cartão. O limite no volume de dados a cada mensagem enviada para o cartão é de 128 *bytes*. Como explicado na Seção 5.1.3, o *IrisCode* é composto por um código de 2048 *bits* e uma máscara de validação também do mesmo tamanho, resultando em um total de 512 *bytes*. Logo, serão necessárias 4 mensagem para que o *IrisCode* seja transferido por completo.

A forma de armazenamento do *IrisCode* foi projetada de modo a facilitar o uso do código em geral e a operação de translação em particular. Esta operação é usada para melhorar a qualidade dos resultados analisados na Seção 5.5.2. Considere a parte *código* de um *IrisCode*, que é uma matriz de 8×256 *bits*, ilustrada na Figura 45a. O Java Card pode armazenar no máximo 16 *bits* em uma única variável do tipo *short*. A forma mais usual de alocação de uma matriz em um vetor é armazenar os elementos linha a linha. Neste caso (Figura 45b), isto consiste em armazenar os *bits* de 1 à 16 da primeira linha na primeira entrada do vetor, os *bits* de 17 à 32 da primeira linha na segunda e assim por diante. Cada linha seria armazenada em um vetor de 16 entradas do tipo *short* ($16 \times 16 = 256$). Portanto, a matriz é armazenada em um vetor de $16 \times 8 = 128$ entradas do tipo *short*.

Para essa implementação foi escolhida uma forma diferente de armazenamento, ilustrada na Figura 45c. Na primeira entrada do vetor são alocadas as duas primeiras colunas (16 *bits*), na segunda, são colocadas as próximas duas colunas e assim por diante, resultando também em um vetor de 128 entradas do tipo *short*. Ambas as formas de alo-



(a) *IrisCode* (b) *Armazenamento por linhas* (c) *Armazenamento por colunas*

Figura 45: Exemplos de imagens de íris do CASIA Iris V4 Interval

cação resultarão no mesmo algoritmo de distância de Hamming uma vez que a distância é feita com o uso da operação XOR bastando apenas que os códigos comparados estejam armazenados da mesma forma. O ganho da forma de alocação proposta está na simplicidade de realizar a translação de *bits*, conforme explicada na Seção 5.2. Cada entrada do vetor estará armazenando exatamente os dados que sofrerão deslocamento. Se o código estivesse armazenado linha a linha, seria necessário fazer várias aplicações da operação deslocamento de *bits*, elevando a complexidade e número de instruções a serem usadas, e em consequência aumentando o tempo de execução.

O Algoritmo 6 detalha os passos usados para implementar a Distância de Hamming, a Equação 5. Sejam n o tamanho do vetor onde a Matriz do código está armazenada, T o *IrisCode* (código binário T_C e máscara T_M) armazenado no cartão e E (E_C e E_M) representa o *IrisCode* do indivíduo que está requisitando a autenticação.

Algoritmo 6 Algoritmo Distância de Hamming entre *IrisCodes*

entrada T e E
saída $resultadoHD$
1: $Nbits = 0$
2: $NbitsTotal = 0$
3: **para** $i := 1 \rightarrow N$ **faça**
4: $xoredC = T_C(i) \oplus E_C(i)$
5: $mascTotal = T_M(i)$ AND $E_M(i)$
6: $xoredC = xoredC$ AND $mascTotal$
7: Conte o número de *bits* iguais a '1' em $xoredC$ e some à variável $Nbits$
8: Conte o número de *bits* iguais a '1' em $mascTotal$ e some à variável $NbitsTotal$
9: **fim para**;
10: $Nbits = 10 \times Nbits$
11: $NbitsTotal = NbitsTotal/10$
12: $resultadoHD = Nbits/NbitsTotal$

A mesma estratégia de multiplicar o dividendo por 10 e dividir o divisor por 10, usada na implementação da biometria da impressão da palma da mão, também foi

utilizada para esta implementação. Com isso o resultado fica sempre entre 0 e 100 sem a necessidade de variáveis do tipo *float* e sem extrapolar o valor máximo permitido em uma variável *short*, que varia de -32768 a 32767.

5.5 Resultados

A extração do *IrisCode* não é o foco deste projeto, por isso, foi utilizada a ferramenta pronta (MASEK et al., 2003). Libor Masek elaborou um extrator e comparador utilizando a ferramenta MATLAB baseado no trabalho (DAUGMAN, 1993). O extrator foi utilizado a fim de gerar os *IrisCodes* para serem comparados no cartão inteligente. Em (BOWYER; HOLLINGSWORTH; FLYNN, 2008), o extrator de Masek foi citado como detentor de bons resultados o que contribuiu para a escolha de sua utilização no projeto desta dissertação.

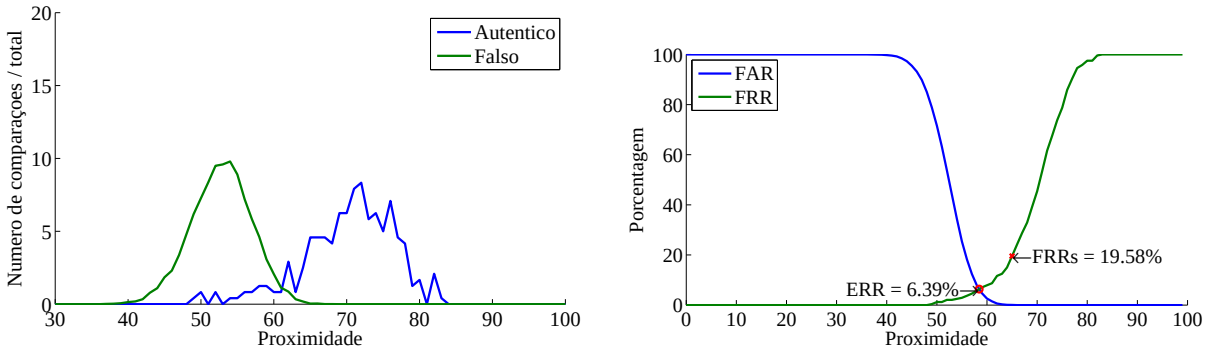
Nesta seção, são apresentados e analisados os resultados das comparações feitas em cartões Java Card usando os bancos de dados CASIA V1 e CASIA V4 Interval. Os resultados serão apresentados relacionando a *proximidade* que é definida por 100 - Distância de Hamming.

5.5.1 Resultados para o CASIA V1

Para o teste foram selecionadas aleatoriamente imagens de 40 diferentes olhos e para cada olho, foram usadas 4 repetições, resultando em 160 imagens. Antes da comparação, os *IrisCodes* foram extraídos utilizando a ferramenta introduzida na Seção 5.1.

A Figura 46 mostra os resultados das $160 \times 160 = 25600$ comparações feitas. A Figura 46a apresenta a distribuição dos resultados das comparações autênticas e falsas em relação aos seus respectivos totais e a Figura 46b mostra o gráfico os valores de FRR e FAR para diferentes valores de proximidade.

O gráfico da distribuição mostra que as comparações autênticas se concentraram em valores de proximidade mais altos enquanto as comparações falsas ficaram concentradas em valores menores. Os gráficos que relacionam os erros à proximidade indicam que o ponto onde os erros são iguais (EER) se aproxima de 6%, mas esse é um valor muito alto para um FAR aceitável, uma vez que um indivíduo diferente do dono do cartão poderia precisar de 20 tentativas para conseguir o acesso. O ponto em que a taxa FAR é menor que 0,1% (FRR seguro) foi escolhido como um limite seguro de proximidade e, para essa

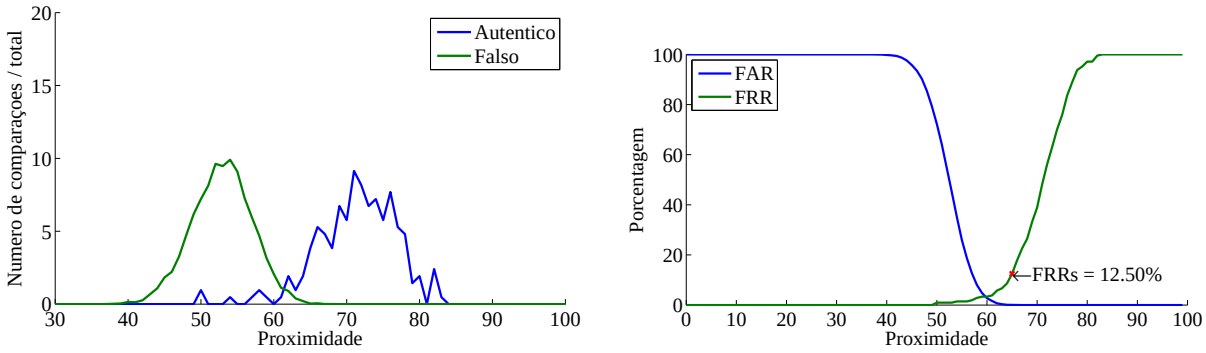


(a) *Distribuição dos resultados* (b) *FAR e FRR*

Figura 46: Resultado das comparações usando o banco CASIA V1

proximidade, o FRR ficou em torno de 20%. Isso significa que o usuário autêntico do cartão teria seu acesso negado uma vez a cada 5 tentativas.

O banco de dados CASIA V1 possui imagens de olhos com pouco desafios (Seção 5.3.1) mas ainda sim é possível que ocorra erros na segmentação das imagens. A Figura 47 mostra os resultados das comparações usando o banco de dados CASIA V1 quando as imagens de íris que ocasionam falhas de segmentação são excluídas. Como esperado, os resultados foram melhores, apresentando uma distribuição de concentrações de comparações autênticas e falsas distintas (Figura 47a). Da mesma forma, o EER diminuiu e o FRR seguro também (Figura 47b).



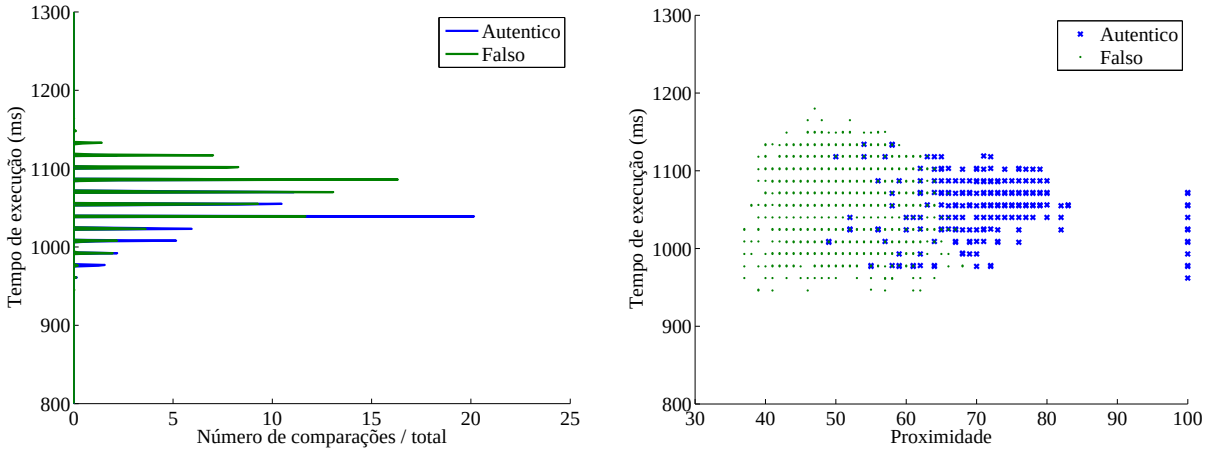
(a) *Distribuição dos resultados* (b) *FAR e FRR*

Figura 47: Resultados sem considerar as falhas de segmentação para o CASIA V1

A Figura 47b mostra o comparativo entre as taxas alcançadas. A partir dos resultados apresentados é possível concluir que a extração tem um grande impacto no processo de comparação pois foi responsável por duplicar a taxa de erro. Apesar de de implicar

um aumento da taxa de erro, as falhas durante a segmentação não podem ser evitadas mas apenas reduzir o seu impacto.

A distribuição do tempo de execução relativo ao total de comparações e em termo da proximidade está ilustrada no gráfico da Figura 48. Observe que o resultado não mostra nenhuma dependência entre o resultado da proximidade e o tempo de execução.



(a) *Tempo de execução* (b) *Proximidade em relação ao tempo*

Figura 48: Tempo de execução para comparações com translação de 1 *bit*

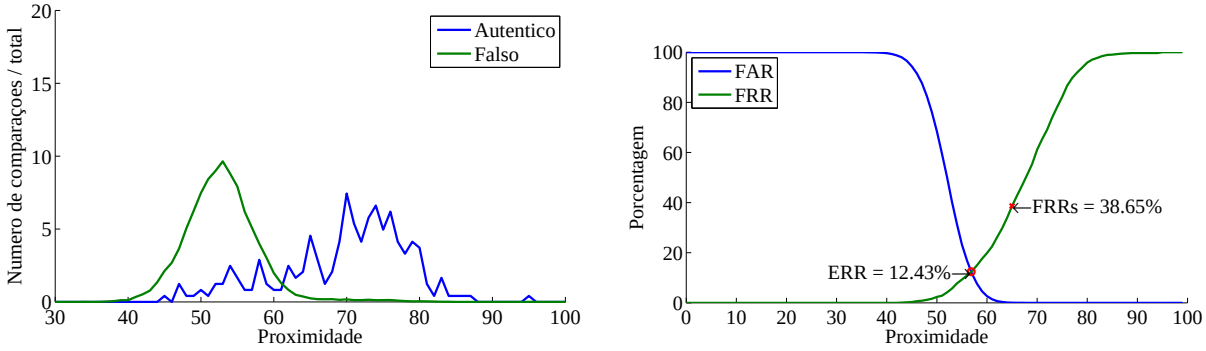
O tempo de execução médio foi de 1052,48 ms e o desvio padrão de 51,93 para comparações autênticas e média 1071,36 ms com desvio padrão 33,06. No início da Seção 5.5 foi explicado que são necessárias 4 trocas de mensagem para que o *IrisCode* seja trasnfarido por completo. Cada troca de mensagem leva 180 ms, totalizando em 720 ms para a transfência do *IrisCode* completo. Aproximadamente 70% do tempo total de execução deve-se a transferência dos dados.

5.5.2 Resultados para o CASIA V4 Interval

O banco de dados CASIA V4 Interval foi introduzido na Seção 5.3.2. Trata-se de um banco de dados com imagens capturadas de forma mais real com problemas de rotação do olho, pouca íris exposta entre outras dificuldades. Foram escolhidas aleatoriamente 200 imagens de diversos olhos e diversas rotações, uma vez que o banco de dados não traz um número de repetições fixo para cada diferente olho. Todos os *IrisCodes* foram extraídos e nenhuma imagem foi desconsiderada.

A Figura 49 mostra os resultados das comparações quanto à proximidade. Se comparado com os resultados das comparações usando o CASIA V1 (Figura 46), nota-se

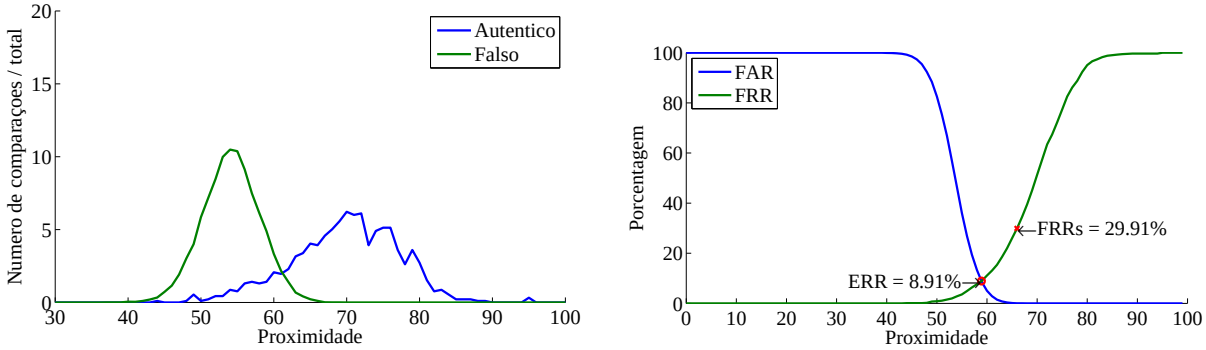
facilmente que o resultado foi muito inferior, ilustrando mais uma vez diferença entre os bancos de dados.



(a) *Distribuição dos resultados* (b) *FAR e FRR*

Figura 49: Resultados para o CASIA V4 Interval

Surge assim a necessidade de aplicar o método que melhore a eficácia do algoritmo de comparação. Na Seção 5.2 foi explicado o método de translação de *bits* aplicado ao *IrisCode*. Usando esse método é possível melhorar os resultados tornando-o assim mais adequado ao uso real. A Figura 50 mostra o resultado das comparações usando a translação de 2 *bits*.

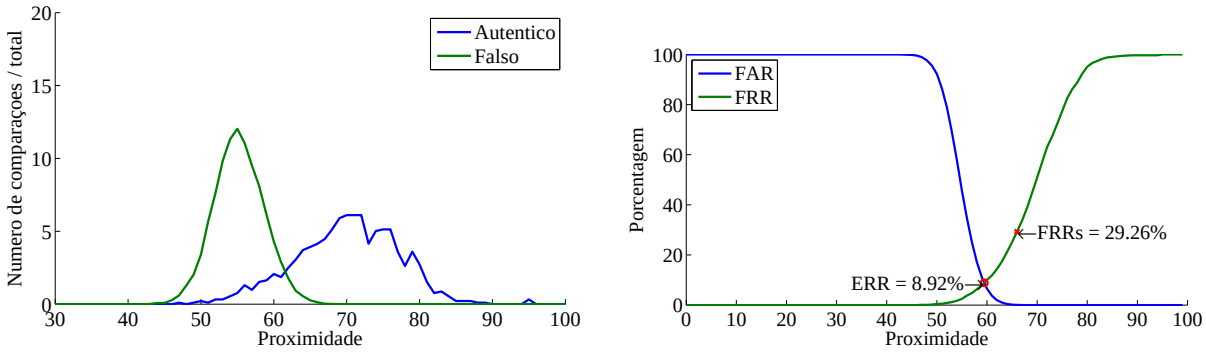


(a) *Distribuição dos resultados* (b) *FAR e FRR*

Figura 50: Resultados CASIA V4 com translação de 2 *bits*

O resultado de FRR seguro teve uma grande redução de 38,65% para 29,91%. Essa melhoria ilustra a necessidade de se usar translações quando há necessidade de garantir eficácia durante as comparações autênticas. A Figura 51 apresenta os resultados das comparações com translação de 4 *bits*.

Nota-se que com a translação de 4 *bits* conseguiu-se uma leve melhoria dos resultados. A melhoria do FRR seguro foi de 29,91% para 29,26%. Na verdade, não parece



(a) *Distribuição dos resultados* (b) *FAR e FRR*

Figura 51: Resultados CASIA V4 com translação de 4 *bits*

muito significativa pois certamente acarreta em um grande aumento do tempo de execução. Afim de analisar os resultados da proximidade e tempo de execução utilizando diferentes translações, foram realizados testes usando translações de 1, 2, 3, 4 e 8 *bits*. A Tabela 9 resume os resultados obtidos com todas as translações consideradas.

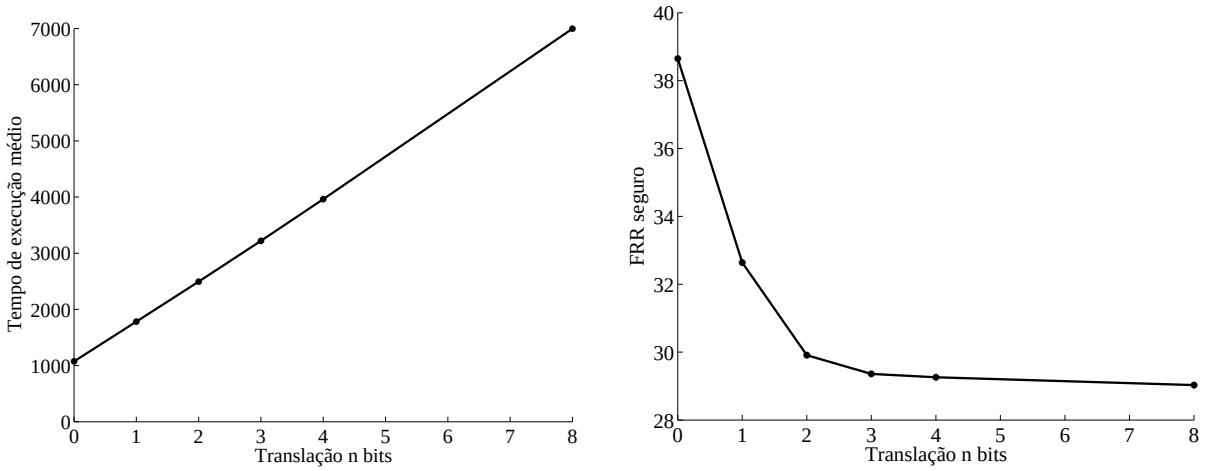
Tabela 9: Resultado das comparações com diferentes translações

Translação	FRR seguro	Tempo médio	Desvio padrão
0	38,65	1074,07	49,84
1	32,64	1782,24	86,92
2	29,91	2495,32	139,73
3	29,36	3220,02	193,78
4	29,26	3963,08	249,67
8	29,03	6998,28	503,63

Os dados da Tabela 9 estão representados na Figura 52. A Figura 52a mostra o gráfico que relaciona o tempo de execução médio usando diferentes translações de *bits*. O gráfico crescente tem inclinação semelhante entre todos os pontos indicando uma relações linear em que soma-se 700 ms para cada translação adicional.

Já a Figura 52b mostra o gráfico que relaciona o FRR seguro à diferentes translações de *bits*. Nota-se que nas translações de 1 e 2 *bits* ocorre uma grande diminuição do FRR seguro mas nas demais translações pouca diferença fazem no resultado.

A partir da translação de 2 *bits*, o resultado do FRR não passa de 29% enquanto o tempo aumenta linearmente. Como na biometria na impressão da palma da mão, é possível melhorar o tempo das comparações autênticas incluindo um limite de aceitação para que o resultado seja antecipado sem que sejam feitas todas as comparações. A Seção 5.5.3 mostra os resultados das comparações usando um limite de aceitação.



(a) *Tempo de execução* (b) *FRR seguro*

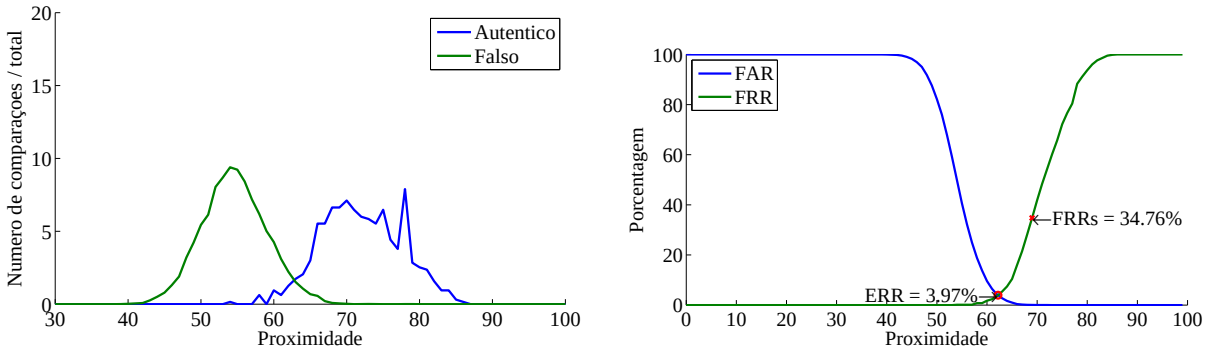
Figura 52: Resultados CASIA V4 com diferentes translações

5.5.3 Resultados de comparações com limite de aceitação

Baseado nos resultados da Seção 5.5.2, foi implementada a comparação com limite da aceitação usando como base a comparação com translação de 2 *bits*. Essa escolha se deve ao tempo de execução e FRR seguro alcançados em comparação com outras translações. A partir da translação de 2 bits, o tempo de execução aumenta mas o FRR seguro continua praticamente o mesmo. O limite foi escolhido como a proximidade de 66 pois para comparações com todas as translações o FAR foi inferior a 0,1%, i.e., a proximidade escolhida foi superior àquela onde ocorre o FRR seguro. Para tornar os testes mais reais e confiáveis, foram usadas 200 novas imagens de olhos do banco de dados CASIA V4 Interval.

A Figura 53 mostra os resultados das comparações. O FRR foi de 34%. No entanto, para a proximidade escolhida como limite de aceitação, o resultado foi um FAR de 0,42% e um FRR de 15,95% devido, principalmente, a mudança do conjunto de imagens analidado. O resultado mostra que a cada 1000 comparações falsas, 4 serão consideradas verdadeiras e a cada 100 comparações autênticas, 16 serão dadas como falsas.

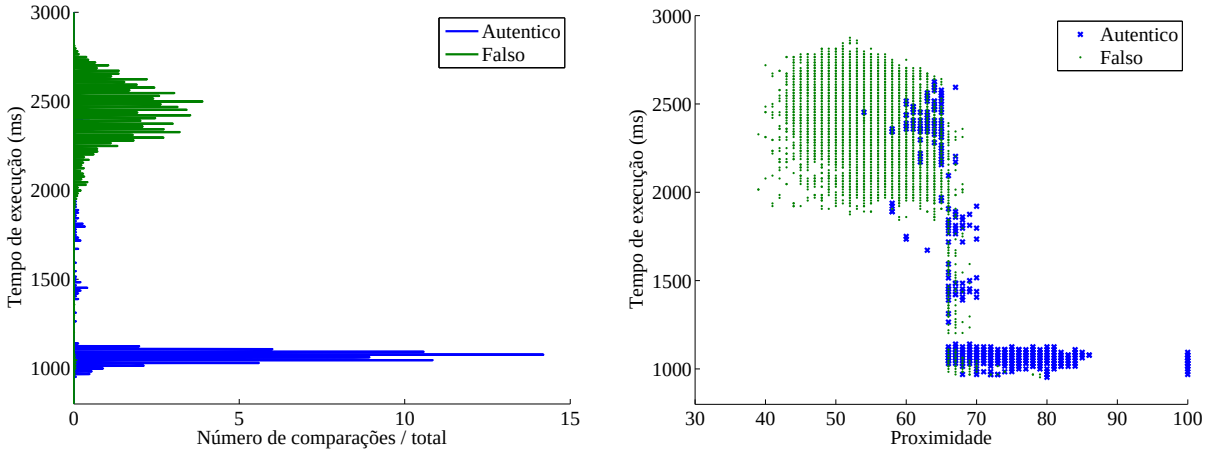
A Figura 54 mostra os tempos de execução, que se benificiaram de um grande ganho com o uso de comparações com limite de aceitação. A média do tempo de execução para as comparações autênticas é de 1210 ms com um desvio padrão de 395, enquanto a média para comparações falsas foi de 2430 ms com um desvio padrão de 189. Os resultados indicam que o método utilizado é capaz de diminuir drasticamente o tempo médio das comparações autênticas. Note que, em um sistema real, este tipo de comparação é provavelmente o mais



(a) *Distribuição dos resultados* (b) *FAR e FRR*

Figura 53: Resultados das comparações com limite de aceitação

requisitado. A Figura 54b ilustra a relação entre o tempo de execução e a proximidade e indica, em conjunto a Figura 54a, que a maior concentração de comparações autênticas não completou o ciclo de translações.

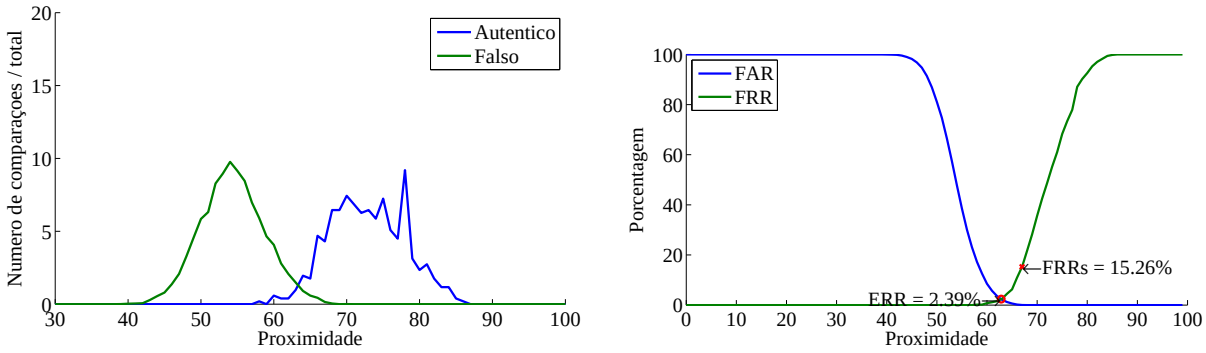


(a) *Tempo de execução* (b) *Proximidade em relação ao tempo*

Figura 54: Tempo de execução para comparações com limite de aceitação

Para entender melhor os resultados da comparação com limite de aceitação levando em consideração apenas os casos de erros, foram excluídas da análise aquelas imagens que não tiveram uma extração satisfatória, como feito anteriormente usando o banco de dados CASIA V1. Os resultados das comparações apenas dos *IrisCodes* de boa qualidade são ilustrado nos gráficos da Figura 55. Estes comprovam que o FRR seguro diminuiu muito em relação ao resultado usando *IrisCodes* com falhas.

Como o método proposto utiliza um limite de aceitação, seria mais correto analisar o resultado com base neste limite pois a execução da verificação sempre é abortada quando



(a) *Distribuição dos resultados* (b) *FAR e FRR*

Figura 55: Resultados das comparações desconsiderando falhas na extração

a proximidade alcança esse limite. Para o limite de aceitação utilizado de proximidade 66, o FAR foi de 0,2% e o FRR 10,96% em contraste com um FAR de 0,42% e FRR de 15,95% obtidos como resultados no caso considerando falhas de segmentação. Houve diminuição tanto em FRR quanto em FAR. Novamente, os resultados mostram a importância do processo de extração pois é capaz de alterar o resultado de forma significativa. Apesar de ser uma biometria desafiadora por possuir tantos obstáculos, o método utilizado se mostra robusto com capacidade de proporcionar bons resultados.

5.6 Considerações Finais

Neste capítulo foram abordadas as diferentes possibilidades de comparação entre *IrisCodes* usando diferentes translações. Provou-se uma boa escolha o uso de um limite de aceitação usando como base a translação de 2 *bits*. Nos testes, os melhores resultados obtidos tiveram seus erros em aproximadamente 10% provando que a biometria da íris é robusta e de fácil comparação. Ficou claro também que um bom resultado de extração é primordial para a eficácia da comparação biométrica.

Como abordado em outros capítulos, a comparação entre biometrias deve ser feita levando em consideração diversos fatores. Os mais significativos para o projeto desta dissertação são o tempo de processamento, memória utilizada e acurácia alcançada. Esta análise é feita no Capítulo 6.

Capítulo 6

COMPARAÇÃO DOS RESULTADOS

NESTE capítulo são comparados os resultados das implementações dos algoritmos de comparação biométricas da impressão digital, íris e impressão da palma da mão que foram detalhadamente apresentadas nos Capítulos 3, 4 e 5, respectivamente. São comparados os desempenhos dos algoritmos a cerca da memória necessária, tempos de execução e acurácia dos resultados.

A Seção 6.1 apresenta as necessidades de memória de cada algoritmo implementado. A Seção 6.3 aborda a comparação sobre o prisma do tempo de execução enquanto a Seção 6.2 trata da acurácia alcançada nos testes das implementações. Uma comparação geral entre as implementações considerando todos esses aspectos é relatada na Seção 6.4.

6.1 Memória

Existem dois tipos de memória em cartões inteligentes: a EEPROM é usada para armazenar os dados do indivíduo possuidor do cartão; e a memória RAM é utilizada para o armazenamento temporário dos dados de entrada para a comparação além de todas as variáveis e estruturas necessárias para a execução do algoritmo de comparação. Note que ambas foram usadas para a implementação dos algoritmos biométricos

A impressão digital é comparada através de suas minúcias e, conforme explicado no Capítulo 3, são usadas no máximo 25 delas para cada impressão. Cada minúcia ocupa 5 bytes, logo, são necessários 125 bytes para garantir o armazenamento completo de uma impressão digital. Uma vez que o número de minúcias pode variar, um byte é necessário para guardar o tamanho da lista de minúcias. Além das minúcias, é necessário armazenar também a tabela de acesso que auxilia durante execução do algoritmo. Essa tabela é

composta por duas colunas de 64 bytes. A soma de todos os bytes necessários é de 254 bytes.

Como visto no Capítulo 4, o espaço necessário para o armazenamento do *PalmCode* é de 256 bytes, sendo que nenhum tipo de dado auxiliar é necessário. Para o armazenamento do *IrisCode*, conforme explicado no Capítulo 5, são necessários 512 bytes, sendo 256 bytes para o código binário e 256 bytes para a máscara.

Os tamanhos da memória de armazenamento, necessária para guardar as impressões digitais e da palma da mão, são praticamente os mesmos sendo 254 e 256 bytes, respectivamente. Já a memória necessária para guardar o *IrisCode* é o dobro (512 Kb). A memória EEPROM disponível no cartão utilizado é de 36 Kb. Logo, os 512 bytes necessários para armazenar o *IrisCode* não representam um problema pois ocupam apenas 1,4% do espaço disponível.

A memória mais crítica no caso de cartões inteligentes é a memória RAM pois, para o cartão utilizado, possui capacidade de armazenamento de apenas 3 Kb e não existem cartões com capacidade de RAM consideravelmente maiores. No caso da implementação da biometria da impressão digital, foi decidido por utilizar um subespaço relativamente grande para conseguir menores tempos de execução. Foi utilizado um subespaço de tamanho 32×32, ou seja, 1 Kb. Além do subespaço é necessário armazenar temporariamente as minúcias de entrada (125 bytes) e algumas variáveis usadas durante a execução que ocupam aproximadamente 128 bytes. Dessa forma, o espaço total necessário para a execução da comparação entre duas impressões digitais é de 1277 bytes.

A implementação da biometria da palma da mão requer um espaço em RAM de apenas 256 bytes para armazenar o *PalmCode* de entrada mais 64 bytes para variáveis auxiliares, totalizando 320 bytes. Similarmente, a RAM requerida para a execução da comparação biométrica da íris é de 512 bytes para armazenar temporariamente o *IrisCode* de entrada mais 64 bytes de variáveis auxiliares, totalizando 576 bytes.

A Tabela 10 mostra os valores dos espaços necessários em memória tanto para armazenamento quanto para alocação temporária durante a execução para as três implementações. Apesar de que a implementação da biometria da íris requerer o dobro do espaço em EEPROM, isso não chega a ser crítico pois ocupa apenas 1,4% do espaço disponível (36 Kb) do cartão utilizado. Vale ressaltar que cartões mais novos podem possuir até 512 Kb, tornando esse número ainda menos expressivo. Por outro lado, o uso da memória

RAM é de extrema importância por não haver muita disponibilidade. Vale ressaltar que nem os cartões mais novo possuem uma RAM muito maior do que 3 Kb.

Tabela 10: Memória necessária

Biometra	EEPROM (bytes)		RAM (bytes)	
Imp. Digital	254	0,68%	1277	41,57%
Imp. Palma da Mão	256	0,69%	320	10,42%
Íris	512	1,38%	576	18,75%

A Figura 56 mostra o gráfico comparativo das memórias necessárias para a execução de cada uma das comparações biométricas.

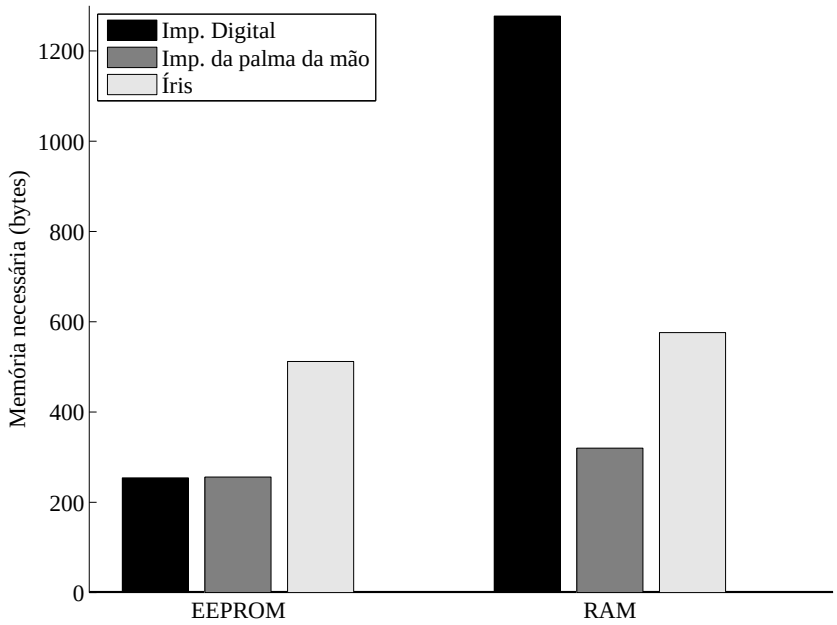


Figura 56: Comparação do tamanho de memória necessário para as biometrias implementadas

Em relação ao uso da RAM, a comparação entre duas impressões digitais chegou a valores significativos ocupando a RAM em aproximadamente 40%. A implementação da biometria da palma da mão e da íris necessitaram de aproximadamente 18% e 10%, respectivamente. Portanto, a implementação da biometria da impressão digital mostrou-se menos adequada que as outras biometrias em relação à esse quesito porém mostrou-se que é possível sua implementação em um cartão inteligente.

6.2 Acurácia

A biometria não se trata de uma simples senha que pode ser comparada para que um acesso seja garantido ou negado. Apesar de atestar a autenticidade de um indivíduo, existem vários fatores que podem levar à uma identificação errada. Esses fatores podem ser dificuldades na extração, nas mudanças no órgão usado para biometria, na captura da imagem de forma ou em ângulos diferentes, entre outros. Portanto a comparação biométrica é dificilmente 100% como seria uma comparação de senhas. Logo, a acurácia indica a chance de acerto no processo de autenticação.

Como visto nas análises das biometrias, um dado muito importante retirado dos testes é o FRR seguro, que nada mais é que o valor do FRR para um limite de aceitação no qual o FAR não ultrapasse 0,1%. Ou seja, valor de limite em que as falsas comparações tem a grande maioria das tentativas negadas.

A Tabela 11 mostra os valores mínimos e máximos de FRR seguros encontrados nos testes feitos para cada biometria. As medidas feitas para a biometria da íris desconsiderando as falhas de segmentação não são aproveitados pois não representam uma situação real de uso do sistema biométrico.

Tabela 11: Acurácia das biometrias

Biometria	#testes	FRR seguro	
		Mínimo	Máximo
Imp. Digital	9	30,77	60,34
Imp. Palma da Mão	5	0	10,44
Íris	7	29,26	38,65

Os resultados dos testes da impressão da palma da mão variaram entre bons (10%) e perfeitos (0%) enquanto os melhores resultados obtidos pelos testes da comparação da íris e da impressão digital ficaram próximos de 30%. Essa grande discrepância entre os resultados é parcialmente explicada pela dificuldade de extração de cada uma das biometrias. A extração do código da íris pode ser muito complicada caso esta esteja coberta pelos cílios ou pálpebras e, por sua vez, a extração da impressão digital pode resultar em um pequeno conjunto de minúcias.

Apesar da possibilidade de alcançar melhores resultados, ignorando esses problemas, não teríamos resultados condizentes com uma situação real, no qual esse processo

é automático e suscetível a erros. Apesar da importância da acurácia da biometria, é necessário também analisar o tempo de execução das comparações.

6.3 Tempo de Execução

O uso das biometrias envolve comparações que servem para permitir ou negar o acesso de um certo indivíduo à aplicações de outros serviços. Logo, o tempo de resposta do sistema deve ser tolerável considerando o tempo de espera desse indivíduo. Portanto, o tempo de execução da comparação biométrica é importante para a aceitação da biometria.

Foram realizados vários testes com variadas parametrizações para as 3 biometrias estudadas. Os testes foram analisados sempre considerando o grupo das comparações autênticas vs. falsas. Nos casos de uso normal de um sistema biométrico em uma cartão digital é esperado que apenas comparações autênticas sejam realizadas, caso contrário, é uma tentativa de transgressão da segurança, tornando sem importância o tempo de execução das comparações falsas. Por esse motivo, apenas as comparações autênticas foram consideradas na análise dos tempos de execução.

A Tabela 12 mostra as menores e maiores médias dos tempos de execução considerando os diferentes testes realizados das comparações biométricas para cada uma das três implementações. Por se tratar de uma comparação visando um contexto real, os tempos de transmissão foram considerados, dessa forma, o tempo pode ser entendido como tempo de espera do usuário.

Tabela 12: Tempo de execução das biometrias

Biometria	#testes	Menores		Maiores	
		Média (ms)	Des. padrão	Média (ms)	Des. padrão
Imp. Digital	9	1372	918	12748	8291
Imp. Palma da mão	5	489	22	3725	109
Íris	7	1074	49	6998	503

Os tempos de execução médios das comparações entre impressões da palma da mão foram os menores seguidos pelas comparações de íris e, por último, as comparações de impressões digitais. O resultado reflete a complexidade de cada um dos algoritmos utilizados. As comparações da íris e palma da mão são similares porém o *IrisCode* é maior e, por isso, tem maior tempo de execução. O algoritmo de comparação entre

impressões digitais possui mais etapas e uma complexidade maior, justificando o maior tempo de execução.

Apesar de importante, o estudo desconexo do tempo de execução não é conclusivo pois apenas varia de acordo com os parâmetros. Na Seção 6.4 serão analisados os tempos de execução e a acurácia em conjunto. Dessa forma, é possível decidir quais algoritmos e parâmetros são ideais para serem usados em um cartão inteligente.

6.4 Considerações Finais

Devido ao baixo poder de processamento do cartão, o tempo de execução se torna crítico. A acurácia é importante para que se possa validar a autenticidade de um indivíduo em tempo real. Portanto, a análise dos resultados deve considerar ambos os fatores. Estes podem variar dependendo de quais parâmetros dos algoritmos são alterados. Já a memória utilizada não varia para esses casos. Logo, a impressão da palma da mão apresenta o melhor resultado neste quesito pois é a que ocupa menos espaço tanto em EEPROM (praticamente o mesmo que a impressão digital) quanto em RAM.

Para que um algoritmo possa ser utilizado em situações reais não basta que ele consiga atingir pequenas taxas de erro, ainda precisa proporcionar isso em tempo de execução reduzido. Para estudar essa relação, foi elaborado o gráfico que relaciona o tempo de execução ao FRR seguro, conforme mostrado na Figura 57. Os melhores resultados são aqueles próximos a origem pois indicam uma porcentagem de erro baixa e um tempo de execução médio curto.

Os resultados obtidos pela comparação biométrica da impressão da palma da mão obtiveram qualidade superior ao da íris e da impressão digital. Apenas os testes biométricos da impressão da palma da mão conseguiram o desejado FRR seguro igual a zero. As comparações da íris obtiveram bons resultados com tempos de execução relativamente baixos. No entanto, os resultados do algoritmo e parametrização dos testes da biometria da impressão digital não alcançaram bons resultados.

Para efeito de comparação à uma situação real, serão considerados adequados em tempo de execução menor do que 3 segundos e um FRR seguro menor do que 40%. Apenas uma das 4 relações obtidos nos testes da impressão da palma da mão ficaram fora do que é considerado adequado. A maioria dos resultados obtidos com os testes da

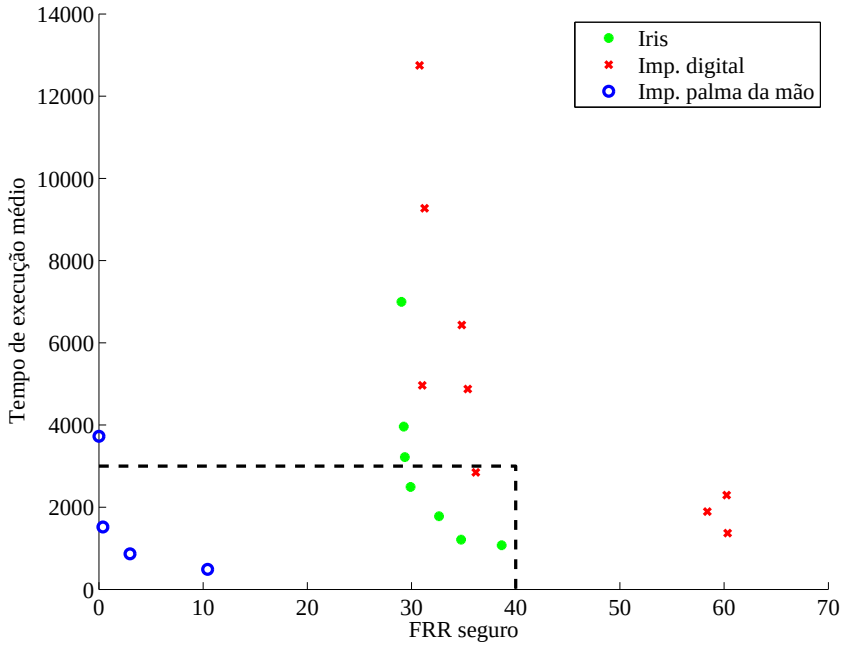


Figura 57: FRR seguro em relação ao tempo de execução

íris conseguiram se enquadrar no limite do considerado adequado. Já para o caso da impressão digital apenas um dos testes conseguiu atingir os limites mínimos.

Essa grande diferença dos resultados pode ser explicada pelos próprios processos de extração e comparação dos algoritmos. O processo biométrico da impressão da palma da mão apresentado foi o mais simples pois possui fácil extração, uma vez que não há obstáculos durante a captura de sua imagem, e permite um processo de comparação simples. Apesar de necessitar um número grande de comparações quando são feitas muitas translações, os resultados já se mostram bons quando usadas uma translação de 1 bit, que é relativamente simples.

O processo de extração das características da íris possui uma dificuldade maior pois normalmente esta está parcialmente obstruída dificultando ou até impossibilitando um processo correto de segmentação para delimitar a íris. Os obstáculos também implicam na necessidade do uso de máscara o que torna maior o espaço necessário para armazenamento assim como o processamento realizado. Como visto no Capítulo 5, os resultados são bem aprimorados quando são desconsideradas as imagens com erro de extração.

A extração da impressão digital também pode ser bem complicada pois possui muitos detalhes e pode gerar minúcias de baixa qualidade. O algoritmo utilizado também precisa da configuração dos limites máximos da translação e rotação. Logo, uma compara-

ção nunca conseguirá uma proximidade alta caso a translação ou rotação real estejam fora dos limites adotados, o que justifica os baixos valores de FRR seguros encontrados, além disso, o processo de comparação é mais complexo pois é dividido em etapas de registro e comparação, o que, inevitavelmente, faz com que o tempo de processamento tenda a aumentar.

O Capítulo 7 tira conclusões quanto a aderência dos resultados em relação ao problema estudado no projeto desta dissertação assim como possíveis melhorias a serem realizadas no futuro.

Capítulo 7

CONCLUSÕES E TRABALHOS FUTUROS

ESTE capítulo apresenta as principais conclusões obtidas pelos resultados dos testes das implementações realizadas neste projeto completando assim a dissertação. Algumas possíveis melhorias são apresentadas para cada uma das implementações assim como possíveis trabalhos futuros a serem realizados utilizando biometrias em conjunto com os cartões inteligentes.

7.1 Conclusões

Esta dissertação analisou a utilização de sistemas biométricos nos quais a comparação final é processada dentro de um cartão inteligente. Foram implementadas três diferentes comparações biométricas utilizando a plataforma Java Card. Antes da implementação, um estudo foi feito para que fosse escolhido um algoritmo adequado para a implementação em cartões inteligentes, dotados de processadores de baixa frequência e pouca memória tanto do tipo EEPROM quanto do tipo RAM disponível.

A comparação biométrica de impressões digitais foi desenvolvida baseada no trabalho (CHOUTA et al., 2012). Trata-se de um algoritmo baseado na comparação de minúcias e tem como foco a diminuição da memória necessária para realização da autenticação. O algoritmo usado para a comparação biométrica da impressão da palma da mão (ZHANG et al., 2003) foi baseado no da íris (DAUGMAN, 1993), e por isso, são muito semelhantes. Ambos fazem uso do filtro 2D de Gabor para extrair um código binário da textura da imagem capturada dos órgãos para então compará-lo utilizando a distância de Hamming. A extração do código da íris (IrisCode) mostrou-se mais complexa pois existe uma difi-

culdade maior em segmentar a imagem para encontrar a região de interesse correta, ou seja, delimitar na imagem exatamente o que é íris e o que não é.

Esse estudo foi motivado pela crescente necessidade de sistemas de segurança cada vez mais robustos. Biometrias são conhecidas como ferramentas de segurança de alta confiabilidade assim como os cartões inteligentes, que foram inseridos em vários segmentos para adicionar novas funcionalidades com segurança. Logo, trata-se de duas tecnologias já ligadas a segurança sendo usadas em conjunto para dar origem a uma nova ferramenta ainda mais robusta.

A tecnologia Java Card foi escolhida tendo em vista a sua vasta documentação disponível, diversas ferramentas e comunidade que puderam auxiliar no processo de desenvolvimento. As três implementações seguiram um fluxo comum: escolha do algoritmo, escolha do banco de dados, implementação do algoritmo, depuração, testes e avaliação dos resultados. Devido à semelhança entre os processos, foi criada uma ferramenta única para auxiliar no processo de implementação, depuração e, principalmente, na realização automática dos testes. Para testar o algoritmo de comparação biométrica da impressão digital foram feitos 9 diferentes testes e, para cada teste, foram realizadas 6400 comparações. Para o caso da íris, foram 7 testes realizando 25600 comparações em cada. Em cada um dos 5 testes da impressão da palma da mão foram realizadas 40000 comparações. A ferramenta desenvolvida foi responsável pelas comparações automáticas realizadas no cartão e tinha como entrada os códigos de cada biometria e o resultado das comparações como saída.

A comparação biométrica da impressão digital implementada é baseada em minúcias. Antes da comparação das minúcias armazenadas com as de entrada é necessário que seja feito um alinhamento entre elas. Foi utilizada a abordagem de subespaços para que a implementação desse algoritmo fosse possível pois sem ela não haveria RAM suficiente no cartão inteligente. Para ajudar no processamento foi utilizada a tabela de acesso que contém índices para as minúcias usando como referência seus ângulos. Como o Java Card não possui bibliotecas matemáticas nem operações com ponto flutuante, foi utilizada uma tabela de consulta para auxiliar no cálculo de operações simples de seno e cosseno. Foram testadas várias parametrizações diferentes e o melhor resultado relacionando tempo de execução médio e o FRR seguro foi alcançado utilizando 16 subespaços e variações de ângulo entre -1 e 1. Nesse caso, o tempo de execução médio foi de 2849 ms e o FRR

seguro foi de 36,17 indicando que o proprietário do cartão teria seu acesso negado uma vez a cada três tentativas, aproximadamente.

A comparação da impressão da palma da mão alcançou os melhores resultados tanto em relação ao tempo de execução quanto à acurácia. A distância de Hamming é calculada entre os *PalmCodes* armazenado e de entrada para obter uma avaliação da proximidade dos dois códigos. Esse resultado pode ser melhorado caso sejam feitos deslocamentos nas direções horizontal e vertical. No caso, utilizando as translações foi possível alcançar um ERR de 0%, i.e., utilizando um limite de aceitação específico, nenhum erro de comparação foi encontrado. em outros termos, todas as comparações falsas foram barradas e todas as comparações verdadeiras foram aceitas. No entanto o uso das translações faz com que o tempo de execução aumente. Para remediar esse problema, um limite de aceitação foi definido para que a execução cessasse assim ele fosse atingido. O resultado é retornado assim que o limite é alcançado. Utilizando essa estratégia foi possível melhorar o tempo de execução médio de 1520 ms para 868 ms em comparações com translação de 1 bit. Para as comparações com translação de 2 bits, que alcançaram a taxa ERR de 0%, o tempo de execução médio foi de 3725 ms sem limite de aceitação e 1217 ms com. O FRR seguro encontrado nas comparações com limite de aceitação e translação de 1 e 2 bits foram praticamente iguais, indicando que o a utilização da translação de 1 bit é suficiente.

Nos testes do algoritmo implementado para a biometria da íris, os resultados não foram tão bons quanto os resultados da biometria da impressão da palma da mão. Para o caso da íris, o melhor resultado alcançado foi um FRR seguro de 34% com um tempo de execução médio de 1210 ms. Para essa biometria também foram testadas translações e assim como foi definido um limite de aceitação à exemplo da biometria da impressão da palma da mão. Grande parte dos erros encontrados se devem à grande dificuldade de extração pois quando as imagens com falhas, que não obtiveram uma extração de qualidade, foram desconsideradas, o FRR seguro alcançou a taxa de 15,26%.

Portanto, em relação ao tempo de execução e à acurácia dos algoritmos escolhidos em conjunto com os bancos de dados utilizados, a biometria da impressão da palma da mão se mostrou ideal pois obteve tempos de comparação inferiores à um segundo e ainda alcançou uma taxa de FRR seguro inferior à 5% podendo esse valor chegar à 0% aumentando um pouco o tempo de execução da comparação. A comparação biométrica da

íris se mostrou robusta e relativamente rápida porém extremamente dependente da etapa de extração obtendo bons resultados quando são desconsiderados os erros de extração. A comparação biométrica da impressão digital obteve os piores resultados pois, em seu melhor teste, alcançou um FRR seguro de apenas 36,17% em conjunto à um alto tempo de execução médio de 2849 ms além de um alto desvio padrão de 1901.

7.2 Trabalhos Futuros

Nesta Seção, são sugeridas algumas formas de continuar o estudo do tema abordado. São muitos os trabalhos possíveis envolvendo biometria e cartões inteligentes. Nesta dissertação, foram desenvolvidas as comparações biométricas da impressão digital, da impressão da palma da mão e da íris todas processadas em cartões. Existem inúmeras biometrias, como a da voz, assinatura, geometria da mão, veias da mão entre outras que podem ser implementadas para que a comparação seja processada em cartões inteligentes. Já existem estudos em que são feitas fusões entre duas biometrias para aumentar a confiabilidade. Algoritmos desse tipo também poderiam ser implementados em cartões inteligentes.

A biometria da impressão da palma da mão foi implementada e alcançou resultados ótimos chegando 0% de erro e um tempo de execução inferior à 1 segundo. Em contra partida, o algoritmo utiliza apenas a extração de uma textura. É indicado que haja um estudo sobre as formas de fraudar a comparação utilizando apenas imagens da mão autênticas e formas de se evitar tais fraudes. Já existe o estudo que relaciona a leitura 3D da palma da mão para evitar certos tipos de fraudes mas existe a possibilidade do uso de processamento de imagens em vídeos evitando a fraude por uma simples imagem. Imagine que o usuário, antes de apresentar a mão, seja requerido a mostrar um número com os dedos para validar que se trata realmente de uma mão e não de apenas uma imagem.

O foco do projeto desta dissertação foi mostrar a viabilidade da implementação de biometrias processadas em cartões inteligentes e comparar as biometrias implementadas em relação à acurácia e ao tempo de execução. Esses fatores são muito importantes mas existem vários outros que podem ser usados em uma comparação mais profunda que validariam a possibilidade da criação de um sistema biométrico completo utilizando cartões inteligentes para o processamento das comparações. São eles: preço dos equipamentos de extração, durabilidade da biometria, unicidade, facilidade de extração, aceitação por parte dos usuários, impacto de doenças degenerativas entre diversos outros fatores.

REFERÊNCIAS

ALLIANCE, S. C. Smart card technology in us healthcare: Frequently asked questions. 2012.

AO, S.; REN, W.; TANG, S. Analysis and reflection on the security of biometrics system. In: IEEE. *Wireless Communications, Networking and Mobile Computing, 2008. WiCOM'08. 4th International Conference on*. [S.l.], 2008. p. 1–5.

ASHBAUGH, D. R. *Quantitative-qualitative friction ridge analysis: an introduction to basic and advanced ridgeology*. [S.l.]: CRC press Boca Raton, 1999.

BOWYER, K. W.; HOLLINGSWORTH, K.; FLYNN, P. J. Image understanding for iris biometrics: A survey. *Computer vision and image understanding*, Elsevier, v. 110, n. 2, p. 281–307, 2008.

BRADLEY, J. N.; BRISLAWN, C. M.; HOPPER, T. Fbi wavelet/scalar quantization standard for gray-scale fingerprint image compression. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. *Optical Engineering and Photonics in Aerospace Sensing*. [S.l.], 1993. p. 293–304.

CAMUS, T. A.; WILDES, R. Reliable and fast eye finding in close-up images. In: IEEE. *Pattern Recognition, 2002. Proceedings. 16th International Conference on*. [S.l.], 2002. v. 1, p. 389–394.

CAPPELLI, R.; MAIO, D.; MALTONI, D. Modelling plastic distortion in fingerprint images. In: *Advances in Pattern Recognition ICAPR 2001*. [S.l.]: Springer, 2001. p. 371–378.

CASIA. *CASIA Iris Database*. [S.l.], 2004. Acessado em outubro de 2012. Disponível em: <http://biometrics.idealtest.org/>.

CEGUERRA, A.; KOPRINSKA, I. Integrating local and global features in automatic fingerprint verification. *Proc ICPR*, v. 3, p. 347–350, 2002.

CHANG, J.; FAN, K. Fingerprint ridge allocation in direct gray-scale domain. *Patt Recog*, v. 34, p. 1907–1925, 2001.

CHANGHONG, L.; ZHAOYANG, L. Efficient iris recognition by computing discriminable textons. *International Conference on Neural Networks and Brain*, v. 2, p. 1164–1167, 2005.

CHEN, Z.; KUO, C. A topology-based matching algorithm for fingerprint authentication. In: IEEE. *Security Technology, 1991. Proceedings. 25th Annual 1991 IEEE International Carnahan Conference on*. [S.l.], 1991. p. 84–87.

CHOU, C. et al. Iris recognition with multi-scale edge-type matching. *International Conference on Pattern Recognition*, p. 545–548, 2006.

CHOUTA, T. et al. A small and high-performance coprocessor for fingerprint match-on-card. In: IEEE. *Digital System Design (DSD), 2012 15th Euromicro Conference on*. [S.l.], 2012. p. 915–922.

COUNCIL, H. *Smart Cards and Biometrics in Healthcare Identity Applications*. [S.l.], 2012.

DAUGMAN, J. High confidence visual recognition of persons by a test of statistical independence. *IEEE Trans. Pattern Anal. Mach. Intell.*, v. 15, p. 1148–1161, 1993.

DAUGMAN, J. Statistical richness of visual phase information. *Int. J. Comput. Vis.*, v. 45, p. 25–38, 2001.

DAUGMAN, J.; DOWNING, K. Epigenetic randomness, complexity and singularity of human iris patterns. *Proc. R. Soc. Lond. B*, v. 268, p. 1737–1740, 2001.

DAUGMAN, J. G. Two-dimensional spectral analysis of cortical receptive field profiles. *Vision research*, Elsevier, v. 20, n. 10, p. 847–856, 1980.

DAUGMAN, J. G. et al. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Optical Society of America, Journal, A: Optics and Image Science*, v. 2, n. 7, p. 1160–1169, 1985.

DU, Y. Using 2-d log-gabor spatial filters for iris recognition. *Biometric Technology for Human Identification III*, 2006.

ECLIPSE. *Eclipse IDE*. [S.l.], 2012. Acessado em maio de 2012. Disponível em: <http://www.eclipse.org/>.

FARINA, A.; VAJNA, Z.; LEONE, A. Fingerprint minutiae extraction from skeletonized binary images. *Patt Recog*, v. 32, p. 877–889, 1999.

FITZ, A.; GREEN, R. Fingerprint classification using a hexagonal fast fourier transform. *Patt Recog*, v. 29, p. 1587–1597, 1996.

GABOR, D. Theory of communication. part 1: The analysis of information. *Electrical Engineers-Part III: Radio and Communication Engineering, Journal of the Institution of*, IET, v. 93, n. 26, p. 429–441, 1946.

GPSHELL. *Global Plataform Shell*. [S.l.], 2012. Acessado em maio de 2012. Disponível em: <http://sourceforge.net/projects/globalplatform/files/>.

HACHEZ, G.; QUISQUATER, J.-J.; KOEUNE, F. Biometrics, access control, smart cards: a not so simple combination. In: *Smart Card Research and Advanced Applications*. [S.l.]: Springer, 2000. p. 273–288.

HAN, C. et al. Personal authentication using palm-print features. *Pattern Recog.*, v. 36, 2003.

HAN, Y.; TAN, T.; SUN, Z. Palmprint recognition based on directional features and graph matching. *Prceedings of the International Conference on Biometrics*, p. 2074–2077, 2007.

HATANO, T. et al. A fingerprint verification algorithm using the differential matching rate. *Proceedings of ICPR*, v. 3, p. 799–802, 2002.

HU, D.; FENG, G.; ZHOU, Z. Two-dimensional locality preserving projection (2dlpp) with its application to palmprint recognition. *Pattern Recog.*, v. 40, p. 339–342, 2007.

HUANG, Y.; DASS, S.; JAIN, K. Localized iris image quality using 2-d wavelets. *International Conference on Biometrics*, v. 1, p. 373–381, 2005.

HUBEL, D. H.; WIESEL, T. N. Ferrier lecture: Functional architecture of macaque monkey visual cortex. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, JSTOR, p. 1–59, 1977.

HUNG, D. Enhancement and feature purification of fingerprint images. *Patt Recog*, v. 26, p. 1661–1671, 1993.

ISENOR, D.; ZAKY, S. Fingerprint identification using graph matching. *Patt Recog*, v. 19, p. 113–122, 1986.

JAIN, A.; CHEN, Y.; DEMIRKUS, M. Pores and ridges: High-resolution fingerprint matching using level 3 features. *IEEE Trans. Pattern Anal.*, p. 15–27, 2007.

JAIN, A.; FENG, J. Latent palmprint matching. *IEEE Trans. Pattern Anal.*, p. 1032–1047, 2009.

JAIN, A.; HONG, L.; BOLLE, R. On-line fingerprint verification. *IEEE Trans Patt Anal Mach Intell*, v. 19, p. 302–314, 1997.

JAIN, A. et al. An identity-authentication system using fingerprints. *Proc IEEE*, v. 85, p. 1365–1388, 1997.

JAIN, A. et al. Filterbank-based fingerprint matching. *IEEE Trans Imag Proc*, v. 5, p. 846–859, 2000.

JAIN, A. K.; ROSS, A.; PRABHAKAR, S. An introduction to biometric recognition. *Circuits and Systems for Video Technology, IEEE Transactions on*, IEEE, v. 14, n. 1, p. 4–20, 2004.

JCDK. *Java Card Development Kit*. [S.l.], 2012. Acessado em maio de 2012. Disponível em: <http://www.oracle.com/technetwork/java/javame/javacard/download/devkit/index.html>.

JCWDE. *Java Card Workstation Development Environment*. [S.l.], 2012. Acessado em maio de 2012. Disponível em: <http://eclipse-jcde.sourceforge.net/user-guide.htm>.

JDK. *Java Development Kit*. [S.l.], 2012. Acessado em maio de 2012. Disponível em: <http://www.oracle.com/technetwork/java/javasebusiness/downloads/java-archive-downloads-javase13-419413.html>.

JING, X.; TANG, Y.; ZHANG, D. A fourier-lda approach for image recognition. *Pattern Recog.*, v. 38, p. 453–457, 2005.

KITTLER, J. et al. On combining classifiers. *IEEE Trans. Pattern Anal. Mach. Intell*, v. 20, p. 226–239, 1998.

KONG, A.; ZHANG, D. Competitive coding scheme for palmprint verification. *Proceedings of the 17th International Conference on Pattern Recognition*, p. 520–523, 2004.

KONG, A.; ZHANG, D.; KAMEL, M. Palmprint identification using feature-level fusion. *Pattern Recog.*, v. 39, p. 478–487, 2006.

KUMAR, A.; ZHANG, D. Personal authentication using multiple palmprint representation. *Pattern Recognition*, v. 38, p. 1695–1704, 2005.

KUMAR, A.; ZHANG, D. Personal recognition using hand shape and texture. *IEEE Trans. Image Process.*, v. 15, p. 2454–2461, 2006.

LEE, C.; WANG, S. Fingerprint feature reduction by principal gabor basis function. *Patt Recog*, v. 34, p. 2245–2248, 2001.

LEE, S.; NAM, B. Fingerprint recognition using wavelet transform and probabilistic neural network. *Proc IJCNN*, v. 5, p. 3576–3279, 1999.

LEE, W.; CHUNG, J. Fingerprint recognition algorithm development using directional information in wavelet transform domain. *Proceedings of the IEEE International Symposium on Circuits and Systems*, p. 1201–1204, 1997.

LI, W.; YOU, J.; ZHANG, D. Texture-based palmprint retrieval using a layered search scheme for personal identification. *IEEE Trans Multimedia*, v. 7, p. 891–898, 2005.

LI, W. et al. Principal line-based alignment refinement for palmprint recognition. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions*, v. 42, p. 1491–1499, Novembro 2012. ISSN 1094-6977.

LI, W.; ZHANG, D.; XU, Z. Palmprint identification by fourier transform. *Int. J. Pattern Recog. Artif. Intell.*, v. 16, p. 417–432, 2002.

LI, Y.; WANG, K.; ZHANG, D. Títpalmprint recognition based on translation zernike moments and modular neural network. *Lecture Notes on Computer Science*, v. 3497, p. 177–182, 2005.

LIU, Y. et al. A practical iris acquisition system and a fast edges locating algorithm in iris recognition. *IEEE Instrumentation and Measurement Technology Conference*, p. 166–168, 2003.

LU, G.; ZHANG, D.; WANG, K. Palmprint recognition using eigenpalms feature. *Pattern Recog. Lett.*, v. 24, p. 1463–1467, 2003.

MA, L. et al. Personal identification based on iris texture analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, v. 25, p. 1519–1533, 2003.

MA, L. et al. Efficient iris recognition by characterizing key local variations. *IEEE Trans. Pattern Anal. Mach. Intell.*, v. 13, p. 739–750, 2004.

MAIO, D.; JAIN, A. K. *Handbook of fingerprint recognition*. [S.l.]: springer, 2009.

MAIO, D.; MALTONI, D. Direct gray-scale minutiae detection in fingerprints. *IEEE Trans Patt Anal Mach Intell*, v. 19, p. 27–40, 1997.

MAIO, D.; MALTONI, D. Neural network based minutiae filtering in fingerprints. *Proc ICPR*, v. 2, p. 1654–1658, 1998.

MAIO, D. et al. Fvc2000: fingerprint verification competition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, IEEE, v. 24, n. 3, p. 402–412, 2002.

MAIO, D. et al. Fvc2002: Second fingerprint verification competition. In: IEEE. *Pattern Recognition, 2002. Proceedings. 16th International Conference on*. [S.l.], 2002. v. 3, p. 811–814.

MAIO, D. et al. Fvc2004: Third fingerprint verification competition. In: *Biometric Authentication*. [S.l.]: Springer, 2004. p. 1–7.

MASEK, L. et al. *Recognition of human iris patterns for biometric identification*. Tese (Doutorado) — Masters thesis University of Western Australia, 2003.

MFCP2. *Mated Fingerprint Card Pairs 2*. [S.l.], 2004. Acessado em julho de 2012. Disponível em: <http://www.nist.gov/srd/nistsd14.cfm>.

NBIS. *NBIS*. [S.l.], 2012. Acessado em março de 2012. Disponível em: <http://www.nist.gov/itl/iad/ig/nbis.cfm>.

NETBEANS. *NetBeans IDE*. [S.l.], 2012. Acessado em maio de 2012. Disponível em: <https://netbeans.org/>.

PHILLIPS, P.; BOWYER, K.; FLYNN, P. Comments on the casia version 1.0 iris data set. *IEEE Trans on Pattern Anal. and Machine Intelligence*, v. 29, n. 10, p. 1869–1870, 2007.

POLYU. *PolyU 3D Palmprint Database*. [S.l.], 2008. Acessado em fevereiro de 2013. Disponível em: <http://www.comp.polyu.edu.hk/ biometrics/2D_3D_Palmprint.htm>.

POON, C.; WONG, D.; SHEN, H. Personal identification and verification: fusion of palmprint representations. *Proceedings of the European Conference on Biometric Authentication*, p. 782–788, 2004.

PRABHAKAR, S.; JAIN, A. Decision-level fusion in fingerprint verification. *Patt Recog*, v. 35, p. 861–874, 2001.

PRABHAKAR, S.; JAIN, A.; PANKANTI, S. Learning fingerprint minutiae location and type. *Patt Recog*, v. 36, p. 1847–1857, 2003.

RAO, A. *A taxonomy for texture description and identification*. [S.l.]: Springer, 1990.

RAO, T. Feature extraction for fingerprint classification. *Patt Recog*, v. 8, p. 181–192, 1976.

RATHA, N.; CHEN, S.; JAIN, A. Adaptative flow orintation based feature extraction in fingerprint images. *Patt Recog*, v. 28, p. 1657–1672, 1995.

RATHA, N. K. et al. A real-time matching system for large fingerprint databases. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, IEEE, v. 18, n. 8, p. 799–813, 1996.

ROSS, A.; REISMAN, J.; JAIN, A. Fingerprint matching using feature space correlation. *Lecture Notes in Computer Science*, v. 2359, p. 48–57, 2002.

SHU, W.; ZHANG, D. Automated personal identification by palmprint. *Optical Engineering*, International Society for Optics and Photonics, v. 37, n. 8, p. 2359–2362, 1998.

STEINBERG, J. Your new iphone can put your identity at risk. *Forbes*, 2013. Acessado em outubro de 2013. Disponível em: <http://www.forbes.com/sites/josephsteinberg/2013/09/13/your-new-iphone-can-put-your-identity-at-risk/>.

SUN, Z.; TAN, T.; QIU, X. A genral framework of iris recognition. *Proc. BioAW Workshop*, p. 270–282, 2004.

SUN, Z. et al. Ordinal palprint representation for personal identification. *Proceedings os the International Conference on Computer Vision and Patter Recognition*, p. 279–284, 2005.

SUNG, H. et al. Iris recognition using collarette boundary localization. *International Conference on Pattern Recognition*, p. 857–860, 2004.

THAI, R. Fingerprint image enhancement and minutiae extraction. *The University of Western Australia*, 2003.

TICO, M. et al. Fingerprint recognition using wavelet features. *Proc ISCAS*, v. 2, p. 21–24, 2001.

WANG, S.; LEE, C. Fingerprint recognition using directional micropattern histograms and lvq networks. *Proc Info Intell Sys*, p. 300–303, 1999.

WANG, X. et al. Palmprint identification using boosting local binary pattern. *Proceedings of the International Conference on Pattern Recognition*, p. 503–506, 2006.

WILDES, R. Iris recognition: An emerging biometric technology. *Proc. IEEE*, v. 85, p. 1348–1363, 1997.

WU, X.; WANG, K.; ZHANG, D. Fuzzy directional element energy feature (fdeef) based palmrpint identification. *Proceedings of the 16th International Conference on Pattern Recognition*, p. 95–98, 2002.

WU, X.; ZHANG, D.; WANG, K. Fisherpalms based palmprint recognition. *Pattern Recog. Lett.*, v. 24, p. 2829–2838, 2003.

WU, X.; ZHANG, D.; WANG, K. Fusion phase and orientation information for palmprint authentication. *Pattern Anal. Appl.*, v. 9, p. 103–111, 2006.

WU, X.; ZHANG, D.; WANG, K. Palm line extraction and matching for personal authentication. *IEEE Transactions on Systems, Man and Cybernetics*, p. 978–987, 2006.

XIAO, Q.; RAAFAT, H. Fingerprint image postprocessing: a combined statistical and structural approach. *Patt Recog*, v. 24, p. 985–992, 1991.

YAGER, N.; AMIN, A. Fingerprint verification based on minutiae features: a review. *Pattern Analysis and Applications*, Springer, v. 7, n. 1, p. 94–113, 2004.

YANG, J. et al. Globally maximizing, locally minimizing: Unsupervised discriminant projection with applications to face and palm biometrics. *IEEE Trans. Pattern Anal. Mach. Intell.*, v. 29, p. 650–664, 2007.

YOU, J. et al. On hierarchical palmprint coding with multiple features for personal identification databases. *IEEE Trans. Circuits Syst. Video Technol*, v. 14, p. 234–243, 2004.

ZHANG, D. et al. Online palmprint identification. *IEEE Trans. Pattern Anal*, p. 1041–1050, 2003.

ZHANG, D. et al. Palmprint recognition using 3-d information. *IEEE trans. Syst. Man Cybern*, p. 505–519, 2009.

ZHANG, D.; ZUO, W.; YUE, F. A comparative study of palmprint recognition algorithms. *ACM Computing Surveys*, v. 44, n. 1, 2012.

ZUO, W.; ZHANG, D.; WANG, K. Bi-directional pca with assembled matrix distance metric for image recognition. *IEEE Trans. Syst. Man Cybern.*, v. 36, p. 863–872, 2006.