

Detecção e Correspondência de Minúcias em Impressões Digitais

J. P. V. Nascimento Filho

H. Pedrini

Relatório Técnico - IC-PFG-19-06

Projeto Final de Graduação

2019 - Julho

UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE COMPUTAÇÃO

The contents of this report are the sole responsibility of the authors.
O conteúdo deste relatório é de única responsabilidade dos autores.

Detecção e Correspondência de Minúcias em Impressões Digitais

José Pedro Vieira do Nascimento Filho* Hélio Pedrini†

Resumo

As principais atividades realizadas durante o Projeto Final de Graduação do curso de Engenharia de Computação do Instituto de Computação da Universidade Estadual de Campinas (UNICAMP) são apresentadas e discutidas neste relatório. O trabalho teve como principal objetivo estudar e desenvolver um detector de minúcias em imagens de impressões digitais e uma ferramenta de correspondência dessas imagens baseada nas minúcias. O método utilizado na detecção é composto de três etapas: (i) etapa de pré-processamento, em que as imagens de entrada passam por algumas técnicas de forma a melhorar sua qualidade; (ii) etapas de afinamento e detecção, em que todas as bifurcações e terminações são encontradas e marcadas; (iii) etapa de pós-processamento, que visa remover falsas minúcias encontradas na fase anterior. O processo de correspondência também foi realizado em três fases: (i) obtenção de uma imagem do banco de dados para comparação de suas minúcias com as da imagem de entrada; (ii) alinhamento da segunda em relação à primeira por meio de seus conjuntos de minúcias; (iii) verificação da correspondência entre as minúcias de cada imagem a fim de encontrar digitais que possam ser do mesmo dedo da mesma pessoa.

1 Introdução

Biometria [1, 2] é o estudo dos fenômenos biológicos, tais como características físicas e comportamentais de um indivíduo, por meio de métodos estatísticos e com o objetivo de reconhecer pessoas unicamente.

Há vários exemplos de identificadores biométricos: retina, íris, DNA, face e impressão digital. No entanto, esta última apresentada pode ser considerada a mais utilizada como identificador atualmente. Há muitos motivos que explicam essa grande utilização da impressão digital, um deles é que enquanto a fabricação de sensores de reconhecimento para digitais possui baixo custo, sensores para outras características biométricas podem apresentar custos elevados. Outro motivo é que a impressão digital pode ter um custo de processamento menor se comparada às outras características biométricas. Sendo assim, há sensores de impressores digitais instalados em vários sistemas diferentes, tais como telefones celulares, caixas eletrônicos, fechaduras de portas, etc.

*Instituto de Computação, Universidade Estadual de Campinas, 13083-852, Campinas, SP, Brasil.

†Instituto de Computação, Universidade Estadual de Campinas, 13083-852, Campinas, SP, Brasil.

As impressões digitais são formadas por cristas papilares e vales. As cristas são as regiões mais acentuadas da pele, enquanto que os vales são as regiões que separam uma crista de outra. Ao coletar uma impressão digital utilizando-se tinta, as linhas pretas formadas representam as cristas, enquanto as linhas brancas os vales.

Um modo de identificação pessoal é por meio do casamento de minúcias encontradas nas impressões digitais. As minúcias são acidentes encontrados em linhas papilares, que fazem com que a crista fuja do seu arranjo natural, vindo a terminar abruptamente ou se bifurcar. Exemplos podem ser vistos na Figura 1.

Representação	Descrição
	Terminação
	Lago
	Bifurcação
	Espora
	Ponto ou Ilha
	Ponte

Figura 1: Exemplos de tipos de minúcias. Adaptado de [13].

Para que haja correspondência entre duas impressões digitais e, então, o reconhecimento da pessoa, é preciso que haja a coincidência de pelo menos dez pontos característicos, que são os acidentes, entre as imagens.

2 Técnicas Relacionadas

Na literatura, há diversas técnicas descritas para detecção de minúcias. Essas técnicas podem ser categorizadas conforme a Figura 2.

Para este trabalho, escolheu-se a técnica que envolve o uso de imagens binárias com afinamento e detecção feita com o auxílio do método *Crossing Number*, além do uso de algoritmos de pré- e pós-processamento.

2.1 Pré-Processamento

É na fase de pré-processamento que a imagem de entrada tem sua qualidade melhorada. O objetivo é segmentar a digital e remover ruídos para que sejam reduzidos os riscos de introdução de falsas minúcias na imagem de saída.

Neste trabalho, utilizou-se um algoritmo escrito em linguagem de programação Python disponível na plataforma *Github* em [10]. O algoritmo está dividido em cinco partes: normalização e segmentação, mapa de orientação, frequência de cristas, filtros de Gabor e binarização.

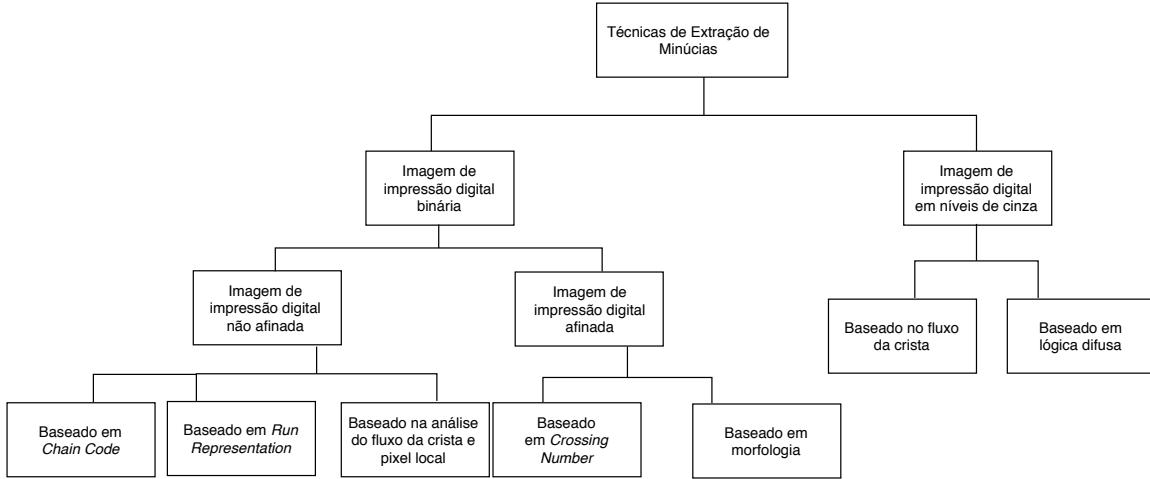


Figura 2: Divisão das técnicas de detecção de minúcias. Adaptado de [14].

2.1.1 Normalização

O processo de normalização consiste em gerar uma distribuição mais uniforme nos níveis de cinza da imagem por meio da utilização de valores escolhidos para a média e a variância, auxiliando, deste modo, a remover os ruídos na imagem. No caso deste projeto, os valores de intensidade da imagem foram normalizados de forma que as regiões de cristas tivessem média zero e desvio padrão unitário.

2.1.2 Segmentação

O processo de segmentação é responsável por dividir uma imagem em várias regiões ou objetos, para que sua análise seja realizada de maneira mais simples. No caso de imagens de impressões digitais, há uma separação entre os pixels que pertencem ao fundo da imagem, o qual está relacionado à área de ruídos nas bordas da imagem, e os pixels que pertencem à impressão digital, originada pelo contato da digital com o sensor.

Para fazer essa separação, o algoritmo divide a imagem de entrada em blocos e calcula o desvio padrão para cada região. Uma região é definida como um conjunto de pixels. Se o desvio padrão estiver acima de um limiar definido, a região é considerada como sendo parte da impressão digital. Caso contrário, ela faz parte do fundo da imagem.

2.1.3 Mapa de Orientação

O mapa de orientação consiste em uma matriz, de mesmo tamanho da imagem de entrada, em que cada elemento representa uma orientação local, estimada no interior de uma vizinhança local, do pixel correspondente na imagem de entrada. A orientação local é definida como sendo o ângulo entre o eixo horizontal e as cristas que atravessam a vizinhança local [12].

A estimativa da orientação é um passo fundamental no processo de enriquecimento da

qualidade da imagem, uma vez que a fase subsequente, de frequência de cristas, depende da orientação local a fim de calcular corretamente a frequência das cristas.

2.1.4 Frequência de Cristas

Esta etapa é responsável por criar uma matriz, representando uma imagem, que contém as frequências das cristas como seus elementos. Esta matriz será usada na fase seguinte de filtros de Gabor para auxiliar na remoção de ruídos.

Imagens de impressões digitais são formadas por cristas e vales. Se bem observadas, é possível perceber um comportamento de uma onda senoidal nas transições entre componentes da digitais, sendo os picos dessa onda representados pelas cristas [12]. Estas estão dispostas de modo paralelo na digital, mas isso não quer dizer que elas estão todas igualmente espaçadas. Por conta dessa variação do espaçamento, há uma frequência relacionada à ocorrência das cristas dentro de um bloco da imagem.

A frequência das cristas é calculada ao longo da direção normal à orientação local da crista dentro de blocos formados a partir da imagem de entrada, ou seja, dentro de uma vizinhança local da qual a crista faz parte.

Caso haja alguma minúcia presente na vizinhança local da crista sendo analisada, a frequência de crista passa a ser definida como a frequência média dos seus vizinhos, pois as minúcias comprometem a formação de uma onda sinusoidal bem definida [12].

2.1.5 Filtros de Gabor

Filtros de Gabor são filtros lineares, sensíveis à orientação e compostos por uma onda sinusoidal em determinada frequência e orientação, modulada por uma Gaussiana. Estes filtros são usados, basicamente, para análises de bordas e texturas, verificando, na imagem, a existência de algum conteúdo que possua uma determinada frequência específica f em direções específicas em uma região em torno do ponto de análise. Desta forma, as cristas e vales são preservados e os ruídos são removidos da imagem [3].

2.1.6 Binarização

Binarização é o processo de conversão de imagens monocromáticas em imagens preto e branco. Para isso um limiar T é escolhido para que pixels com valores acima de T sejam marcados como branco, enquanto aqueles que estão abaixo ou iguais a T sejam marcados como preto. Como o filtro de Gabor destacou a diferença entre cristas e vales, ou seja, separou a imagem em duas classes, uma delas sendo representada pela região mais escura (cristas) e a outra pela região mais clara (vales) [11], a binarização é aplicada de modo a evidenciar essa separação.

2.2 Afinamento

Afinamento ou esqueletização de uma imagem analisada consiste no processo de reduzi-la a uma cadeia simples, com a largura de apenas um pixel. Em imagens de impressões digitais, o afinamento é realizado utilizando-se imagens binárias. A qualidade destas tem

influência significativa e direta na imagem final afinada, uma vez que falsas minúcias podem ser introduzidas nesta última. Deste modo, em imagens de impressões digitais, as linhas que formam as cristas são reduzidas através da remoção de pixels redundantes que as formam até que tenham um pixel de largura.

O algoritmo de afinamento utilizado neste trabalho foi o de Zheng e Suen [6]. Ele opera em imagens binárias e apenas nos pixels pretos que possuem uma 8-vizinhança completa, considerando que as cristas são representadas por eles. Isto significa que os pixels que fazem parte da borda e dos cantos da imagem não são analisados.

A Figura 3 ilustra uma 8-vizinhança de um pixel central preto P_1 sendo analisado.

P_9	P_2	P_3
P_8	P_1	P_4
P_7	P_6	P_5

Figura 3: Pixel preto e sua vizinhança

Seja $A(P_1)$ o número de transições zero-para-um (branco-para-preto) na sequência $P_2 \rightarrow P_3 \rightarrow P_4 \rightarrow P_5 \rightarrow P_6 \rightarrow P_7 \rightarrow P_8 \rightarrow P_9 \rightarrow P_2$. $B(P_1)$ é o número de vizinhos de P_1 que são pretos.

O algoritmo opera em dois passos aproximadamente iguais que são repetidos até que os pixels da imagem parem de mudar, ou seja, até que pixels pretos parem de ser transformados em pixels brancos. Ambos os passos possuem cinco condições que devem ser cumpridas para que o pixel preto analisado passe a ser um pixel branco, sendo as duas últimas diferentes em cada passo. No primeiro passo, o pixel analisado passa pelas seguintes condições:

1. P_1 é preto e possui oito vizinhos;
2. O número de pixels vizinhos de P_1 que são pretos é no mínimo 2 e não é maior que 6. Isto é, $2 \leq B(P_1) \leq 6$;
3. O número de transições do tipo branco-para-preto ao redor de P_1 é igual a 1. Isto é, $A(P_1) = 1$;
4. Pelo menos um dos pixels P_2 , P_4 ou P_6 é branco (igual a 0);
5. Pelo menos um dos pixels P_4 , P_6 ou P_8 é branco (igual a 0).

Já no segundo passo, as condições são as seguintes:

1. P_1 é preto e possui oito vizinhos;
2. O número de pixels vizinhos de P_1 que são pretos é no mínimo 2 e não é maior que 6. Isto é, $2 \leq B(P_1) \leq 6$;
3. O número de transições do tipo branco-para-preto ao redor de P_1 é igual a 1. Isto é, $A(P_1) = 1$;

4. Pelo menos um dos pixels P_2 , P_4 ou P_8 é branco (igual a 0);
5. Pelo menos um dos pixels P_2 , P_6 ou P_8 é branco (igual a 0).

Os pixels que satisfazem todas as cinco condições em cada passo, somente serão transformados em brancos depois que todos os pixels da imagem tenham sido analisados e todos pixels candidatos tenham sido localizados. Maiores detalhes do funcionamento do algoritmo podem ser encontrados em [7].

2.3 Detecção de Minúcias

É nesta etapa que as minúcias são encontradas e marcadas. Como mencionado no início desta seção, a técnica escolhida para detectar minúcias foi a de *Crossing Number*, explicada a seguir.

2.3.1 Crossing Number

Crossing number é a técnica mais utilizada no processo de detecção de minúcias em imagens binárias que foram afinadas por conta da sua eficiência computacional e simplicidade. Utilizando o esqueleto da imagem binária, este método faz uma análise, com o auxílio de uma janela 3×3 pixels, como mostra a Figura 4, de cada pixel 8-conectado e presente nas cristas, e de sua vizinhança para que as minúcias sejam encontradas [5].

P_4	P_3	P_2
P_5	P	P_1
P_6	P_7	P_8

Figura 4: Vizinhança 3×3 pixels.

O valor CN é calculado usando a Equação 1:

$$CN = \frac{1}{2} \sum_{i=1}^8 |P_i - P_{i+1}| \quad (1)$$

em que $P_1 = P_9$. O valor CN é definido como sendo a metade da soma da diferença entre os pares de pixels adjacentes em uma 8-vizinhança. A Tabela 1 mostra a classificação de cada minúcia de acordo com o valor de CN calculado.

2.4 Falsas Minúcias

Várias minúcias são encontradas pelo algoritmo *Crossing Number*, no entanto, muitas delas são falsas. As falsas minúcias, representadas na Figura 5, podem ser introduzidas na imagem através de fatores como ruídos na imagem, e artefatos de imagem criados pelo processo de afinamento. Pelo fato das falsas minúcias não serem algo esperado na imagem da impressão digital, elas precisam ser removidas utilizando algoritmos de pós-processamento.

CN	Tipo de Minúcia
0	Ponto isolado
1	Ponto final
2	Ponto contínuo
3	Ponto bifurcado
4	Ponto cruzamento

Tabela 1: Propriedades do *Crossing Number*.

Representação	Descrição
	Escada
	Lago
	Quebra de Crista
	Espora
	Ponto ou Ilha
	Ponte
	Triângulo
	Fusão

Figura 5: Exemplos de falsas minúcias.

2.5 Pós-Processamento

A fase de pós-processamento é responsável por remover as falsas minúcias dentre todas as que foram encontradas durante a detecção. A remoção é necessária visto que as falsas minúcias podem comprometer o processo de correspondência de imagens que será descrito posteriormente.

Há alguns algoritmos de pós-processamento descritos na literatura, como o baseado em *Fuzzy Rules* [9]. Entretanto, o escolhido para ser implementado neste trabalho foi o descrito por Tico e Kuosmanen [8]. O Algoritmo 1 testa a validade de cada minúcia encontrada na etapa de detecção por meio da análise de uma vizinhança $W \times W$, na imagem afinada, ao redor do pixel em que a minúcia foi detectada.

1 **início**

```

2   Crie uma matriz  $L$  de tamanho  $W \times W$  preenchida com zeros de modo que ela
    represente uma vizinhança  $W \times W$  da imagem afinada  $I$  centrada no pixel da
    minúcia candidata. Isto é, todo elemento de  $L$  está mapeado em um pixel de  $I$ 
    que está no interior da vizinhança de tamanho  $W \times W$  cujo centro é a minúcia
    analisada;
3   Como mostrado nas Figuras 6(a) e 7(a), marque com -1 o pixel central de  $L$ , o
    qual representa a minúcia analisada de  $I$ ;
4   se minúcia é terminação então
5     Marque, em  $L$ , com 1, todos os pixels que estão conectados a terminação
        candidata, mas que são pretos em  $I$  e estão marcados como 0 em  $L$ , como na
        Figura 6(b);
6     . Conte o número de transições zero-para-um existentes ao se realizar um
        percurso no sentido horário entre os pixels que pertencem a borda da imagem
         $L$ . Essas transições são denotadas  $T_{01}$ , como mostrado na Figura 6(c);
7     se  $T_{01} = 1$  então
8       | Minúcia analisada é confirmada como sendo uma verdadeira terminação;
9     fim
10   fim
11   se minúcia é bifurcação então
12     Marque, em  $L$ , com 1, 2 e 3, respectivamente, três componentes conexas
        encontradas durante um percurso no sentido horário da 8-vizinhança de
        pixels da bifurcação candidata. A Figura 7(a) ilustra este passo;
13   para cada  $l = 1,2,3$  faça
14     Marque, em  $L$ , com  $l$  todos pixels que:
15       i. estão marcados como 0 em  $L$ ;
16       ii. estão conectados com o pixel marcado com  $l$ ;
17       iii. correspondem a um pixel preto na imagem afinada de entrada.
18   fim
19   Conte o número de transições zero-para-um, zero-para-dois e zero-para-três
        existentes ao se realizar um percurso no sentido horário entre os pixels que
        pertencem à borda da imagem  $L$ . Essas transições são denotadas  $T_{01}$ ,  $T_{02}$  e
         $T_{03}$ , respectivamente, como mostrado na Figura 7(f);
20   se  $T_{01} = 1$ ,  $T_{02} = 1$  e  $T_{03} = 1$  então
21     | Minúcia analisada é confirmada como sendo uma verdadeira bifurcação;
22   fim
23   fim
24 fim

```

Algoritmo 1: Algoritmo de pós-processamento descrito em [8].

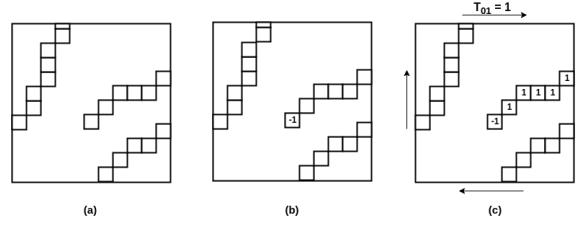


Figura 6: Exemplos de validação de terminação. Adaptado de [8].

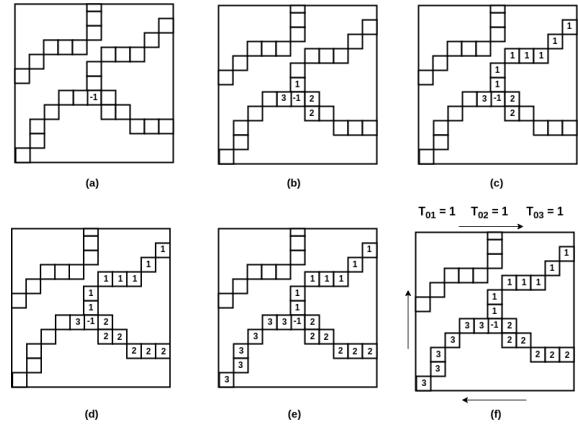


Figura 7: Exemplos de validação de bifurcação. Adaptado de [8].

A dimensão W da vizinhança analisada ao redor de cada minúcia candidata é dada como sendo duas vezes maior que a distância média entre duas cristas vizinhas.

Embora o algoritmo consiga remover falsas minúcias, ele é capaz de detectar e eliminar apenas as que estão associadas com esporas, lagos, pontes, escadas e ilhas. Exemplos de remoção de minúcias podem ser vistos na Figura 8.

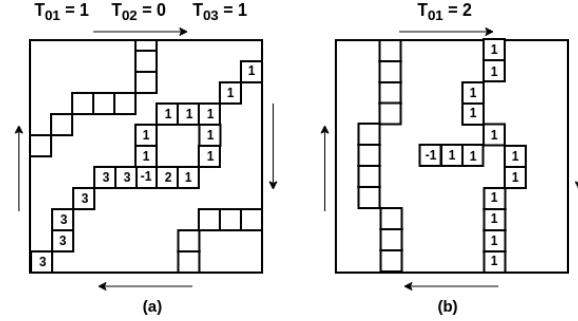


Figura 8: Exemplos de remoção de falsas minúcias: lago (a) e espora (b). Adaptado de [8].

2.6 Correspondência de Digitais

O casamento de digitais consiste em encontrar, dentro de um banco de dados, uma imagem de impressão digital que tenha pontos característicos iguais ou parecidos com os da imagem de digital de entrada, de modo a identificar uma pessoa.

No entanto, muitas vezes, ao coletar a impressão digital de uma pessoa, o posicionamento do dedo não é o mesmo que o da última vez que a impressão foi coletada, o que ocasiona um deslocamento entre as minúcias. Deste modo, é preciso realizar um alinhamento entre os conjuntos de minúcias das impressões digitais sendo analisadas, para que seja possível verificar um possível casamento entre as minúcias dos dois conjuntos, e então validar ou não a correspondência.

O algoritmo apresentado a seguir, descrito em [15], segue esta linha. O casamento é dividido em três fases:

- Registro.
- Pareamento de minúcias.
- Cálculo do número de minúcias casadas.

Cada uma delas será apresentada mais detalhadamente nas próximas subseções.

2.6.1 Registro

Em processamento de imagens, o registro tem como objetivo realizar um alinhamento geométrico entre duas imagens que foram capturadas com algum aspecto diferente, seja ele a data da captura ou o tipo de equipamento. O alinhamento é realizado aplicando-se transformações geométricas, como a da Equação 2. Nela estão representadas as transformações de escala (s), rotação (θ) e translação ($\Delta x, \Delta y$), as quais foram utilizadas para o alinhamento dos conjuntos de minúcias, isto é, mapear o conjunto de minúcias da imagem de entrada no conjunto de imagens do banco de dados.

$$F_{s,\theta,\Delta x,\Delta y} \begin{pmatrix} x \\ y \end{pmatrix} = s \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} \quad (2)$$

Seja E o conjunto de minúcias da imagem de entrada e A o conjunto de minúcias da imagem do banco de dados sendo analisada. Cada minúcia é representada por uma tupla que contém suas coordenadas e sua orientação local.

Na fase de registro do algoritmo, primeiramente, são escolhidos intervalos para o ângulo θ e para o fator de escala s . Entretanto, como o sensor utilizado para coletar as digitais é o mesmo para todas as imagens, o fator de escala foi tratado como sendo igual a 1 neste trabalho. Já para o intervalo de ângulo foi escolhido como sendo $[-90, 90]$.

Para cada minúcia pertencente a E , realiza-se a tentativa de mapeá-la a alguma minúcia de A por meio de alguma transformação geométrica. Para isso, primeiro, encontra-se o ângulo θ do intervalo que ao rotacionar o ponto característico analisado da imagem de entrada usando este ângulo, obtém-se a mesma orientação de algum ponto característico da

imagem do banco de dados. O ângulo θ é descoberto de modo que a soma do ângulo da orientação local da minúcia de E e de θ seja igual ao ângulo da orientação local da minúcia de A .

Caso alguma rotação seja encontrada, calcula-se os valores Δx e Δy da translação por meio da Equação 3, em que e_x, e_y e a_x, a_y são coordenadas das minúcias de E e A , respectivamente.

$$\begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} a_x \\ a_y \end{pmatrix} - s \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} e_x \\ e_y \end{pmatrix} \quad (3)$$

Deste modo, tem-se uma transformação geométrica capaz de alinhar as imagens sendo analisadas. Então, ela é aplicada a todas as minúcias de E para que seja possível calcular a quantidade de minúcias de E que casam com alguma minúcia correspondente de A . O valor calculado é salvo em uma lista juntamente com o valores de θ , Δx e Δy . Após todas as transformações encontradas, a que tiver o maior valor é a escolhida para ser usada na etapa seguinte de correspondência.

2.6.2 Correspondência de Minúcias

A correspondência consiste em aplicar a transformação encontrada, na fase anterior, nas minúcias pertencentes ao conjunto E para que elas formem pares com as do conjunto A , caso possuam mesma orientação e mesmas coordenadas.

Entretanto, para que na fase seguinte não seja preciso verificar toda as minúcias de ambos os conjuntos e assim diminuir a quantidade de processamento computacional, criou-se uma fronteira comum imaginária entre os dois conjuntos de minúcias que englobe o maior número de minúcias dos dois conjuntos. Para isso, primeiramente é traçada uma fronteira individual para cada imagem analisada. Esta precisa englobar todas as minúcias presentes em cada um dos conjuntos. Depois, a intersecção entre as duas fronteiras é calculada para obter a fronteira comum. Assim, apenas as minúcias dentro dessa fronteira comum serão analisadas na fase seguinte do algoritmo.

2.6.3 Cálculo do Número de Minúcias Casadas

Tendo as minúcias do conjunto E já casadas com as do conjunto A e a fronteira comum definida, o algoritmo descreve a criação de vizinhanças de tolerância ao redor de cada minúcia do conjunto E presente na fronteira comum, uma vez que nem sempre as minúcias casadas possuem exatamente as mesmas coordenadas. Essas vizinhanças estão representadas na Figura 9 e o tamanho delas é calculado de acordo com a distância entre cristas e com a distância do núcleo da digital.

Com as vizinhanças de tolerância traçadas, o próximo passo é contar o número de pontos característicos que realmente casaram, ou seja, possuem a mesma orientação e estão dentro da mesma vizinhança de tolerância, o que quer dizer que possuem a mesmas coordenadas. A Figura 9 mostra exemplos de correspondências.

Seja pm o resultado da contagem do número de minúcias que casaram, qE e qA a quantidade de minúcias da imagem de entrada e da imagem do banco de dados presente na fronteira comum, respectivamente. Esse resultado é obtido utilizando-se a Equação 4 para

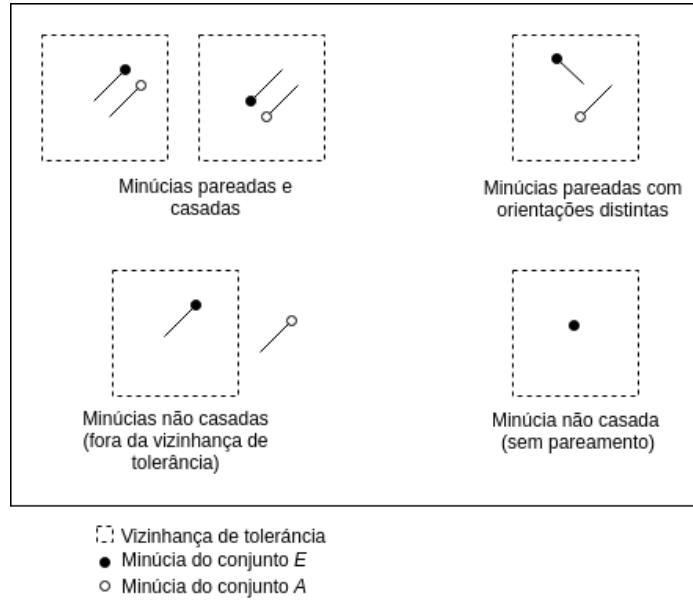


Figura 9: Exemplo de possíveis correspondências. Adaptado de [15]

calcular o valor MS que representa o grau de casamento. Quanto maior for o MS , mais as imagens analisadas são semelhantes.

$$MS = \frac{pm * pm}{qE * qA} \quad (4)$$

Após o cálculo, o valor de MS é guardado em uma lista que contém os dez maiores valores de MS já calculados, isto é, as dez imagens do banco que mais se aproximaram da imagem de entrada.

3 Métodos Propostos

Na realização deste trabalho, foram implementados códigos na linguagem de programação Python em sua versão 2.7. Para os testes, foram utilizadas imagens em níveis de cinza, no formato TIFF e com dimensões 300×300 pixels. Elas podem ser encontradas nos bancos de dados tipo B da FVC 2000 (*Fingerprint Verification Competition*) [4]. Há quatro bancos de dados: DB1, DB2, DB3 e DB4, cada um contendo 80 imagens de impressões digitais, sendo imagens de 10 dedos diferentes e 8 imagens de cada um desses 10 dedos. As imagens de cada banco foram adquiridas por meio de diferentes tipos de dispositivos, como listado abaixo.

- DB1: sensor óptico de baixo custo *Secure Desktop Scanner* por *KeyTronic*.
- DB2: sensor capacitivo de baixo custo *TouchChip* por *ST Microelectronics*.
- DB3: sensor óptico DF-90 por *Identicator Technology*.

- DB4: geração de impressões digitais sintéticas.

Para que fosse possível detectar as minúcias, primeiramente a imagem de entrada passou pela fase de pré-processamento. Nela, a imagem foi normalizada, segmentada, teve seu mapa de orientação e frequência de cristas extraídos, foi filtrada pelos filtros de Gabor e, por fim, sofreu uma binarização, para, então, ser salva.

A imagem obtida no pré-processamento foi afinada com o auxílio do algoritmo descrito para que o método *Crossing Number* fosse aplicado. Todas as minúcias foram encontradas pelo algoritmo e depois marcadas na imagem de saída, sendo os círculos azuis correspondendo às terminações e os círculos vermelhos correspondendo às bifurcações. Esta imagem de saída também foi salva.

Como todas as minúcias existentes na imagem foram encontradas pelo *Crossing Number*, houve a necessidade de pós-processá-la com o objetivo de eliminar as falsas minúcias. Para isso, primeiro eliminou-se as consideradas minúcias de borda da seguinte maneira: para toda minúcia do tipo terminação, duas linhas imaginárias, uma na horizontal e outra na vertical, foram traçadas a partir do pixel que representa a minúcia. Caso, em alguma direção, seja ela direita, esquerda, para cima ou para baixo, só forem encontrados pixels brancos, quer dizer que esta é uma minúcia de borda e pode ser eliminada, uma vez que ela foi criada por conta do sensor.

Depois, foi a vez de remover as falsas minúcias que sobraram. Para isso, o algoritmo de pós-processamento descrito anteriormente foi aplicado. Após a conclusão do algoritmo, a imagem pós-processada foi salva.

Com relação ao casamento, primeiramente uma imagem do banco de dados foi escolhida como base para que fosse possível encontrar outra, presente neste mesmo banco, que possuísse pontos característicos semelhantes e, assim, casasse com ela. A imagem base passou pelo detector de minúcias e teve seu conjunto de minúcias guardado.

Depois, foi a vez de cada uma das imagens presentes no banco ter seu conjunto de minúcias detectado. Após a detecção, utilizando o algoritmo descrito anteriormente, a imagem de entrada foi alinhada em relação à outra imagem, tendo a primeira, seu conjunto de minúcias mapeado ao conjunto da segunda por meio da melhor transformação geométrica encontrada, a qual envolvia transformações de rotação e translação.

Em seguida, uma fronteira comum foi determinada entre o conjunto de minúcias transformado da imagem de entrada e o conjunto da segunda imagem, para que o número de minúcias a ser analisado diminuísse na etapa seguinte de casamento e, consequentemente, diminuísse o processamento computacional.

Por último, foram traçadas as vizinhanças de tolerância ao redor de cada minúcia presente no conjunto de minúcias transformadas, para que, então, fosse contabilizado o número total de casamentos que ocorreram.

O número total de casamentos foi passado para a Equação 4 com o intuito de obter o grau de casamento entre as imagens. O valor deste grau foi salvo em uma lista contendo os dez melhores valores calculados para ele. Ao término da análise de todas as imagens do banco, a lista foi analisada para verificar se alguma imagem da mesma digital estava presente nela e, assim, validar o casamento.

4 Resultados Experimentais

Nesta seção, serão apresentados e discutidos os resultados obtidos pelo algoritmo de detecção de minúcias implementado por meio das imagens de saídas obtidas. Em cada etapa dele, uma imagem de saída foi obtida para cada imagem de entrada, que são elas: 101_1.tif, 102_1.tif, 103_1.tif, 104_1.tif, 105_1.tif, 106_1.tif, 107_1.tif, 108_1.tif, 109_1.tif e 110_1.tif.

Na imagem de entrada, ilustrada na Figura 10(a), pode-se perceber que algumas áreas da impressão digital apresentam falhas ou deformações, como nas regiões marcadas por círculos vermelhos na Figura 10(b). Essas falhas ou deformações podem levar ao surgimento de falsas minúcias, como quebras de crista, lagos, escadas, entre outras.

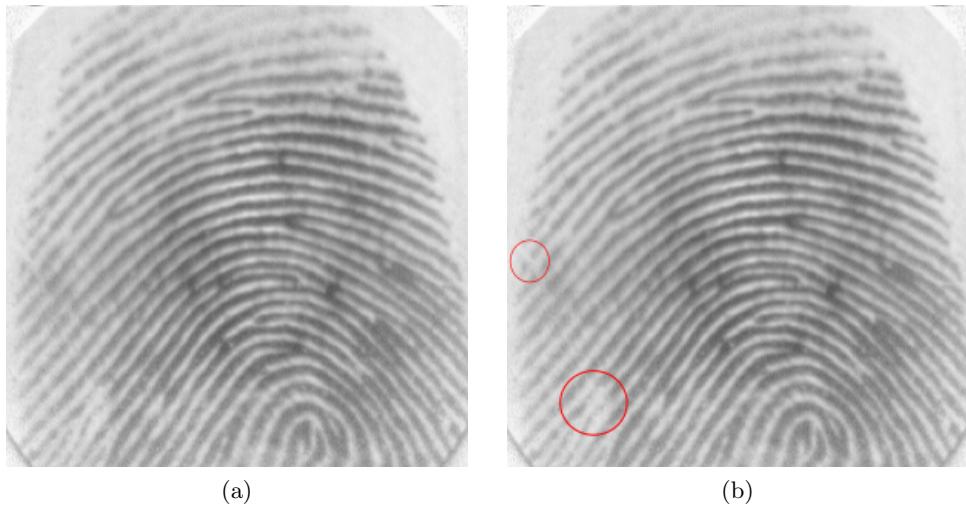


Figura 10: Exemplos de imagem de entrada: Imagem 101_1 do banco de dados DB1.

Há também as falhas que estão relacionadas à pequena pressão exercida pelo dedo sobre o sensor que coleta a impressão digital, causando, assim, o surgimento de ilhas ou pontos e quebras de cristas durante a fase de pré-processamento. Na Figura 11, esse tipo de falha é bastante perceptível, uma vez que é possível enxergar que as linhas que representam as cristas existem, no entanto, possuem uma cor mais clara se comparadas com a da Figura 10(a), com alguns pontos mais escuros em destaque, o que pode levar o algoritmo de pré-processamento a confundir essas cristas com o fundo da imagem.

As Figuras 12(a) e 12(b) ilustram imagens de saída da etapa de pré-processamento. Elas comprovam que as falhas das imagens de entrada podem ser propagadas ao longo do processo de detecção. Deste modo, é notável que as falhas apresentadas nas Figuras 10(b) e 11 são refletidas em suas imagens resultantes do pré-processamento ilustradas nas Figuras 12(a) e 12(b), respectivamente. Na primeira imagem, a falha apontada pelo círculo vermelho menor deu origem a uma quebra de crista, enquanto a falha apontada pelo círculo vermelho maior causou um afinamento ao longo da crista. Já na segunda, as áreas mais claras da imagem de entrada acabaram sendo confundidas com o fundo e os pontos mais escuros presentes nelas deram origem às diversas ilhass e quebras de cristas que se encontram no lado esquerdo da imagem de saída.

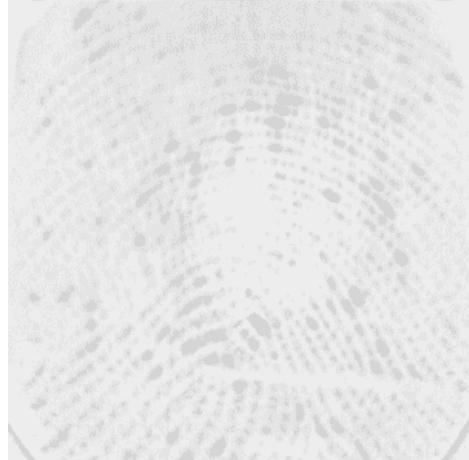


Figura 11: Exemplo de falha por pouca pressão no sensor: Imagem 110_1 do banco de dados DB1.



Figura 12: Exemplos de imagens pré-processadas: Imagens 101_1 e 110_1 do banco de dados DB1.

As imagens retornadas pelo pré-processamento são encaminhadas para a fase de alinhamento. Fazendo uma análise da Figura 13, resultado do afinamento da Figura 10(a), pode-se perceber que o algoritmo de Zheng e Suen escolhido para o afinamento não acrescenta nenhum tipo de artefato de imagem, como falsas minúcias ou falsas conexões entre cristas, à imagem de saída, como pode ocorrer com outros algoritmos. Sendo assim, vê-se que ela é uma cópia fidedigna da imagem de entrada, mas afinada. Tendo-se uma impressão digital afinada de boa qualidade, o processo segue, então, para a fase de detecção com o algoritmo *Crossing Number*.

Como resultado do *Crossing Number*, tem-se a Figura 14. Verificando atentamente essa



Figura 13: Exemplo de imagem afindada: Imagem 101_1 do banco de dados DB1.

figura, pode-se observar que realmente todas as minúcias presentes na imagem afinada foram detectadas, inclusive as falsas e as que foram adicionadas ao longo das etapas anteriores. Isso pode ser comprovado por meio da segunda coluna da Tabela 2, a qual mostra que o número de minúcias encontradas foi superior ao número máximo que uma imagem de impressão digital de boa qualidade pode ter, que é de 80 [14]. Então, para eliminar as minúcias indesejadas, a impressão digital passa pela etapa de pós-processamento.

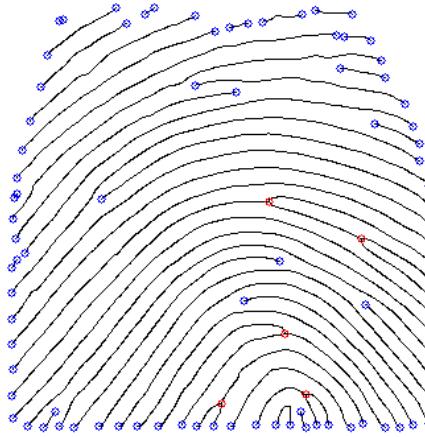


Figura 14: Exemplo de imagem após *Crossing Number*: Imagem 101_1 do banco de dados DB1.

Após o pós-processamento, tem-se como resultado as imagens mostradas na Figura 15. Comparando as imagens obtidas após o *Crossing Number* (Figura 14) e o pós-processamento (Figura 15(a)), observa-se que houve uma diminuição considerável no número de minúcias marcadas. A terceira coluna da Tabela 2 comprova essa observação. Fazendo uma análise desta coluna, pode-se ver que os resultados apresentados foram satisfatórios para a maioria

das imagens, pois, segundo Thakkar [14], uma imagem de digital de boa qualidade pode ter entre 25 e 80 minúcias.

Observando as imagens que ficaram fora desse intervalo sugerido em [14], tem-se que, pela Figura 15(a), que representa a impressão digital que ficou com um número menor do que 25 minúcias, todas as minúcias detectadas são verdadeiras. Já as impressões digitais que apresentaram um número superior a 80 minúcias, ainda existe muitas minúcias que são falsas e não foram removidas pelo pós-processamento, como pode ser visto nas Figuras 15(b) e 15(c).

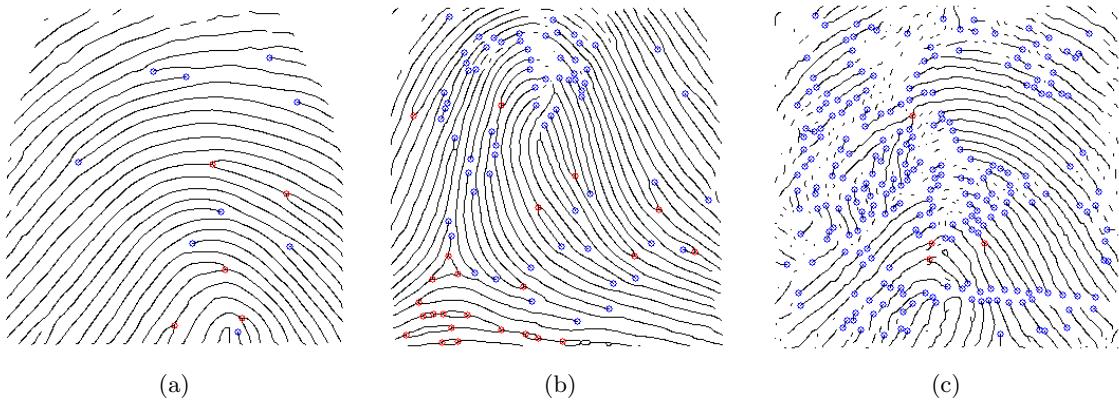


Figura 15: Exemplos de imagem após pós-processamento: Imagens 101_1, 109_1 e 110_1 do banco de dados DB1.

Imagen	# Minúcias Encontradas pelo CN	# Minúcias Após Pós-Processamento
101_1.tif	92	14
102_1.tif	118	50
103_1.tif	152	69
104_1.tif	222	89
105_1.tif	154	61
106_1.tif	132	44
107_1.tif	226	49
108_1.tif	212	76
109_1.tif	216	92
110_1.tif	722	253

Tabela 2: Número de minúcias encontradas após o *Crossing Number* e após o pós-processamento para cada imagem de entrada.

Com relação aos resultados obtidos na etapa de correspondência, apresentados na Tabela 3, pode-se observar que, das dez imagens de entrada usadas no teste, seis delas obtiveram sucesso no casamento, já que o casamento é válido caso uma imagem da mesma impressão digital que a da imagem de entrada apareça na lista que contém os dez maiores

valores para o grau de casamento.

Entrada	Top 1	Top 2	Top 3	Top 4	Top 5	Top 6	Top 7	Top 8	Top 9	Top 10
101_1	101_8	106_1	104_7	107_3	101_3	108_4	107_4	101_4	104_2	103_6
102_1	101_1	106_5	104_7	102_5	107_3	105_7	109_3	105_4	107_8	106_3
103_1	109_6	107_8	101_3	103_3	101_8	103_6	101_4	107_6	102_8	110_7
104_1	101_1	105_6	102_2	101_2	107_6	101_5	109_8	103_6	101_3	102_5
105_1	101_2	101_3	101_5	102_6	105_7	106_6	106_7	103_6	109_4	103_3
106_1	101_1	110_8	107_4	101_4	107_8	109_2	104_3	106_5	108_4	105_7
107_1	106_5	104_2	101_2	101_1	103_2	104_8	110_1	101_5	105_4	101_6
108_1	105_3	105_8	110_2	109_8	107_4	110_3	103_3	101_6	102_7	102_2
109_1	101_8	105_2	107_8	101_1	106_4	109_7	105_5	102_4	104_3	106_8
110_1	105_5	101_4	104_2	102_3	103_8	103_2	101_7	103_6	105_6	102_1

Tabela 3: Resultado do algoritmo de casamento implementado.

As entradas 104_1, 107_1, 108_1 e 110_1 foram as que falharam, segundo a Tabela 3. Analisando as imagens de saída do pós-processamento, observa-se que, em imagens que pertencem à mesma impressão digital, o conjunto de minúcias encontrado para cada uma delas diverge bastante, isto é, há minúcias encontradas que são exclusivas de apenas algumas das imagens, o que pode levar a imagem de entrada a se parecer mais com imagens de outra impressão digital do que com as da mesma impressão digital, como é o caso da impressão digital 107_1, que casou com várias imagens do grupo da impressão digital 101_1, segundo a Tabela 3.

Os fatores que podem ter causado essa divergência no conjunto de minúcias são: a baixa qualidade das imagens obtidas pelo sensor, que afetou as fases de pré-processamento e afinamento, e a não remoção de alguns tipos de falsas minúcias pelo algoritmo de pós-processamento.

5 Conclusões e Trabalhos Futuros

A partir dos resultados obtidos neste trabalho, pode-se concluir que o algoritmo implementado cumpre a sua função de detectar minúcias de imagens de impressões digitais. No entanto, imagens de digitais com baixa qualidade acabam afetando o algoritmo ao introduzir várias minúcias falsas. Por conta disso, as técnicas de pré- e pós-processamento necessitam ser aperfeiçoadas para que menos minúcias falsas apareçam na imagem final.

Com relação ao pré-processamento, algum método mais robusto pode ser usado para complementar o que já está implementado de forma que as ilhas (ou pontos) e as quebras de minúcias possam ser evitadas. Já com relação ao pós-processamento, também é preciso implementar algumas regras que complementem o algoritmo utilizado para que ele seja capaz de remover os tipos de minúcias falsas, uma vez que ele é limitado à remoção de apenas aquelas associadas com esporas, lagos, pontes, escadas e ilhas.

Já o algoritmo implementado para o casamento não está de acordo com o resultado

esperado, ou seja, ter sucesso em todos os casamentos, como apresentado em [15]. Como propostas para trabalhos futuros, um estudo mais aprofundado sobre o assunto, juntamente com refinamentos no pré- e pós-processamento poderão contribuir para que um resultado melhor seja alcançado.

Referências

- [1] Jain, Anil K., Patrick Flynn, Arun A. Ross (2007). Handbook of Biometrics. Springer Science & Business Media.
- [2] Bolle, Ruud M., Jonathan H. Connell, Sharath Pankanti, Nalini K. Ratha, Andrew W. Senior (2013). Guide to Biometrics. Springer Science & Business Media.
- [3] Joshi, Prateek (2014). Understanding Gabor Filters. <https://prateekvjoshi.com/2014/04/26/understanding-gabor-filters/>.
- [4] First International Competition for Fingerprint Verification Algorithms. <http://bias.csr.unibo.it/fvc2000/>.
- [5] Bansal, Roli & Sehgal, Priti & Bedi, Punam. (2011). Minutiae Extraction from Fingerprint Images - A Review. International Journal of Computer Science Issues. 8.
- [6] Zheng, T. Y. and Suen, C. Y. (1984). A Fast Parallel Algorithm for Thinning Digital Patterns. Communications of the ACM, 27, 3 (March), 236-239.
- [7] The Zheng-Suen Thinning Algotithm: Introduction and Applications. <http://jaredgerschler.blog/2018/02/25/the-zhang-suen-thinning-algorithm-introduction-and-applications/>.
- [8] Tico, Marius & Kuosmanen, Pauli. (2000). An algorithm for fingerprint image post-processing. 2. 1735 - 1739 vol.2. 10.1109/ACSSC.2000.911285.
- [9] Stephen, Meka James. (2013). Removal of False Minutiae with Modified Fuzzy Rules. International Journal of Computing. Volume 3. 178-183.
- [10] Deshmukh, U. Fingerprint-Enhancement-Python. <https://github.com/Utkarsh-Deshmukh/Fingerprint-Enhancement-Python>.
- [11] Carreira, R. M. P. C. B. and Gomes, D. M. (2012). Análise e Comparação de Métodos de Pré-Processamento de Imagens e Extração de Características em Impressões Digitais.
- [12] Castro, T. S. & Cerqueira, A. S. and Gouvêa, David S. A. (2008). Identificação de Impressões Digitais Baseada na Extração de Minúcias.
- [13] Jampour, Mahdi & Yaghoobi, Mahdi & Ashourzadeh, Maryam & Soleimani, Adel. (2010). A New Fast Technique for Fingerprint Identification with Fractal and Chaos Game Theory. Fractals-Complex Geometry Patterns and Scaling in Nature and Society - Fractals. 18. 10.1142/S0218348X10005020.

- [14] Thakkar, D. Minutiae Based Extraction in Fingerprint Recognition. <https://www.bayometric.com/minutiae-based-extraction-fingerprint-recognition/>, 2019.
- [15] N. K. Ratha, K. Karu, Shaoyun Chen and A. K. Jain. 1996. A Real-Time Matching System for Large Fingerprint Databases. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, no. 8, pp. 799-813.