



Universidade Federal de Campina Grande - UFCG  
Centro de Engenharia Elétrica e Informática - CEEI  
Departamento de Engenharia Elétrica - DEE

Arthur Cruz de Araújo

## **Relatório de Estágio Integrado**

Campina Grande, Paraíba

Abril de 2017



Universidade Federal de Campina Grande - UFCG  
Centro de Engenharia Elétrica e Informática - CEEI  
Departamento de Engenharia Elétrica - DEE

Arthur Cruz de Araújo

## **Relatório de Estágio Integrado**

Relatório de Estágio Integrado submetido à Unidade Acadêmica de Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciências no Domínio da Engenharia Elétrica.

Orientador:  
Professor Edmar Candeia Gurjão, Dr. Sc.

Campina Grande, Paraíba  
Abril de 2017



Universidade Federal de Campina Grande - UFCG  
Centro de Engenharia Elétrica e Informática - CEEI  
Departamento de Engenharia Elétrica - DEE

Arthur Cruz de Araújo

## **Relatório de Estágio Integrado**

Relatório de Estágio Integrado submetido à Unidade Acadêmica de Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciências no Domínio da Engenharia Elétrica.

Aprovado em        /        /

---

**Professor Edmar Candeia Gurjão, Dr. Sc.**  
Universidade Federal de Campina Grande  
Orientador

---

**Professora Luciana Ribeiro Veloso, Dr. Sc.**  
Universidade Federal de Campina Grande  
Convidada

Campina Grande, Paraíba  
Abril de 2017



Este trabalho é dedicado a meus pais.



# Agradecimentos

Agradeço a minha família pelo suporte desde sempre, possibilitando minha dedicação exclusiva à graduação, além do constante acompanhamento ao longo da mesma, especialmente nos períodos de afastamento para intercâmbio.

Agradeço a minha namorada Carla pelo apoio contínuo, companheirismo e carinho.

Agradeço ao Professor Dr. Edmar Candeia Gurjão, o primeiro professor com quem tive o prazer de trabalhar, no primeiro ano de graduação, ainda nos tempos do PET Elétrica, e que agora, nos últimos meses dela, aceitou me acompanhar como orientador deste estágio.

Agradeço a meu primo Fábio, que me ofereceu orientação e ajuda externa nas atividades que realizei durante o estágio.

Agradeço ao amigo e colega de trabalho Rajiv Albino pela recomendação dentro da Vsoft. A ele, assim como aos amigos de trabalho Arnaldo, João, Iron, Rodrigo e Ramon agradeço pelos momentos de orientação, trocas de ideias e descontração.

Agradeço aos funcionários da Vsoft pelo ambiente de trabalho amigável e a Pedro Alves pela oportunidade e confiança ao aceitar-me como estagiário.



*“Todo pintor pinta a si mesmo.”*

*Michelangelo*



# Resumo

Este relatório descreve as atividades realizadas por Arthur Cruz de Araújo, graduando em Engenharia Elétrica pela Universidade Federal de Campina Grande, durante o estágio curricular integrado de 660 horas, na empresa Vsoft Tecnologia. A empresa vem se especializando no desenvolvimento de *software* voltado ao processamento digital de imagens, estando seu foco principal voltado para a biometria de face e impressões digitais. A classificação de impressões digitais consiste no problema de associar uma dada impressão a uma classe de maneira consistente e confiável. O principal foco do estágio foi a implementação de um algoritmo estrutural para classificação de impressões digitais baseado no casamento inexato de grafos. As atividades desenvolvidas correspondem aos subsistemas do algoritmo implementado: este passa pela geração da imagem de orientação de uma impressão digital, pela sua segmentação em regiões regulares e a posterior criação de um grafo representativo. A partir daí, havendo sido criados grafos protótipos para as classes de interesse, o casamento inexato de grafos é executado, gerando uma medida de distância entre os grafos que possibilita a classificação. As atividades foram desenvolvidas na linguagem C++, utilizando a biblioteca de visão computacional *OpenCV*.

**Palavras-chave:** processamento digital de imagens, otimização, biometria, impressão digital, casamento de grafos.



# Abstract

This paper reports the activities developed by Arthur Cruz de Araújo, an Electrical Engineering undergraduate student at Federal University of Campina Grande, during his integrated 660 hours long internship at Vsoft Tecnologia. The company has been specializing in the development of software related to digital image processing, focusing on face and fingerprint biometrics. The classification of digital fingerprints can be defined as the problem of assigning a given fingerprint to a class in a consistent and reliable manner. The main focus during this internship was the implementation of a structural algorithm for fingerprint classification based on inexact graph matching. The developed activities correspond to the algorithm's subsystems: it passes through the generation of a fingerprint's orientation image, then moves on to its segmentation into regular regions and the posterior creation of a representative graph. Prototype graphs for the classes of interest are created and, using them, the inexact graph matching is performed, generating a measure of distance between the graphs which would allow classification to happen. The activities were developed in C++ language, using the computer vision library OpenCV.

**Keywords:** image processing, optimization, biometrics, fingerprint, graph matching.



# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>21</b>
<b>2</b>	<b>OBJETIVO</b>	<b>23</b>
<b>3</b>	<b>A EMPRESA</b>	<b>25</b>
<b>4</b>	<b>FUNDAMENTAÇÃO</b>	<b>27</b>
<b>4.1</b>	<b>Sistemas biométricos</b>	<b>27</b>
<b>4.2</b>	<b>Erros em sistemas biométricos</b>	<b>27</b>
4.2.1	Erros no módulo de casamento	28
4.2.2	Taxas de erro de verificação	28
<b>4.3</b>	<b>Representação de uma impressão digital</b>	<b>30</b>
4.3.1	Orientação local de cristas e a Imagem de Orientação	31
<b>4.4</b>	<b>Esquema de Classificação Galton-Henry</b>	<b>32</b>
<b>5</b>	<b>ATIVIDADES REALIZADAS</b>	<b>35</b>
<b>5.1</b>	<b>Visão geral do trabalho implementado</b>	<b>35</b>
<b>5.2</b>	<b>Obtenção da imagem de orientação</b>	<b>35</b>
<b>5.3</b>	<b>Segmentação da imagem de orientação</b>	<b>38</b>
<b>5.4</b>	<b>Construção do grafo conexo</b>	<b>41</b>
<b>5.5</b>	<b>Definição de grafos protótipos de referência</b>	<b>43</b>
<b>5.6</b>	<b>Casamento inexato de grafos</b>	<b>45</b>
5.6.1	Definição	45
5.6.2	Operações básicas sobre grafos	46
5.6.3	Busca em espaço de estados pelo casamento ótimo	47
5.6.4	Cálculo do limiar inferior de Gilmore Lawler	50
5.6.5	Um exemplo	54
<b>5.7</b>	<b>Classificação</b>	<b>55</b>
<b>6</b>	<b>CONCLUSÃO</b>	<b>57</b>
	<b>REFERÊNCIAS</b>	<b>59</b>



# Lista de figuras

Figura 1 – Logo VSoft Tecnologia® . . . . .	25
Figura 2 – Um exemplo de curvas $FMR(t)$ e $FNMR(t)$ , onde são marcados os pontos de $EER$ , $ZeroFNMR$ e $ZeroFMR$ . . . . .	29
Figura 3 – Pontos típicos de operação para diferentes aplicações. . . . .	30
Figura 4 – Cristas e vales em uma impressão digital. . . . .	31
Figura 5 – Uma imagem de impressão combinada com sua correspondente imagem de orientação, calculada sobre uma malha de tamanho 16x16. . . . .	32
Figura 6 – As cinco classes de impressões digitais comumente utilizadas. Dois tipos de impressões whorl são mostradas (plain e twin). . . . .	32
Figura 7 – Os passos principais na abordagem estrutural trabalhada. Resultados exemplificam a classificação de uma impressão do tipo <i>left loop</i> . . . . .	36
Figura 8 – Obtenção da imagem de orientação para duas impressões digitais: na primeira linha, uma impressão do tipo arco; na segunda, uma do tipo presilha esquerda (left loop). Ao longo de cada linha, tem-se: (a) imagem original, imagens gradiente no eixo $x$ (b) e $y$ (c), imagem de orientação por pixel (d), imagem de orientação por bloco (e) e imagem de orientação por bloco suavizada (f). . . . .	38
Figura 9 – Pseudo-código do algoritmo de segmentação e do procedimento de reatribuição. . . . .	40
Figura 10 – Exemplo de segmentação para uma impressão do tipo (i) arco, (ii) presilha esquerda e (iii) presilha direita: (a) impressão original, (b) imagem de orientação, (c) resultado da segmentação espiral, (d) resultado final após varredura de TV e remoção de regiões pequenas. Custos totais das segmentações foram, respectivamente: 6135, 15843 e 16936. . . . .	42
Figura 11 – Exemplos dos grafos associados às segmentações previamente apresentadas. Da esquerda para a direita, tem-se um arco, uma presilha esquerda e uma presilha direita. . . . .	43
Figura 12 – Procedimento para criação de grafos protótipos. . . . .	44
Figura 13 – Protótipos criados para as classes: arco (i), <i>left loop</i> (ii), <i>right loop</i> (iii) e <i>whorl</i> (iv). . . . .	44
Figura 14 – Exemplo de um cenário de casamento entre dois grafos $G_1$ e $G_2$ com 3 e 4 nós, respectivamente. . . . .	48
Figura 15 – Resultado do casamento entre os grafos da Figura 14. Custo total do casamento foi de $C(G_1, G_2) = 60138$ . . . . .	49
Figura 16 – Diagrama ilustrando a avaliação do primeiro passo na busca utilizando a abordagem branch-and-bound. . . . .	49

Figura 17 – Casamento de um grafo oriundo de uma classe tipo arco com os protótipos das classes: (i) arco, (ii) presilha esquerda (left loop), (iii) presilha direita (right loop) e (iv) verticilo (whorl). No lado esquerdo, os grafos são apresentados em seus estados iniciais e no lado direito após os respectivos casamentos. . . . .	54
Figura 18 – Matriz de confusão resultante do primeiro teste. . . . .	55

# Lista de abreviaturas e siglas

FVC	Fingerprint Verification Competition
FMR	False Match Rate
FNMR	False Non-Match Rate
EER	Equal-Error Rate
<i>igm</i>	Inexact Graph Matching
LB	Lower Bound
QAP	QUadratic Assignment Problem
LAP	Linear Assignment Problem
GLB	Gilmore Lawler Bound



# 1 Introdução

Reconhecimento biométrico ou, simplesmente, biometria se refere ao uso de identificadores ou características distintivas de natureza anatômica ou comportamental, tais como impressões digitais, face, íris e voz, para o reconhecimento automático de um indivíduo. O reconhecimento pela biometria fornece maior segurança, eficiência e, em muitos casos, maior conveniência para o usuário, o que vem motivando aplicações de caráter tanto governamental (identificação nacional, cruzamento de fronteira, passaportes eletrônicos) quanto civil (logon de usuário em redes, celulares e acesso *Web*). (Maltoni et al., 2009)

O reconhecimento de impressões digitais, em particular, tem-se desenvolvido em uma ampla variedade de áreas, de ciência forense a aparelhos celulares. Curiosamente, embora a impressão digital seja popularmente tida como sinônimo de individualidade, a unicidade de impressões digitais não é fato, mas uma observação empírica. (Maltoni et al., 2009)

Uma boa representação de uma impressão digital deve apresentar *saliência*, isto é, conter informações discriminatórias suficientes, e *adequação*, podendo ser facilmente extraída, armazenada e utilizada para reconhecimento. Representações baseadas em imagem, onde informação é retirada da intensidade dos pixels, são sujeitas a fatores como variações de brilho, qualidade da imagem, marcas e distorções. No entanto, retêm a maior quantidade de informação da impressão.

O reconhecimento de impressões digitais envolve, muitas vezes, sua comparação com um banco de dados extenso. O processo de identificação pode ser acelerado pela redução no número de comparações necessárias. Dados como sexo, raça e idade podem reduzir significativamente o banco efetivo de busca. Entretanto, como tais informações nem sempre estão disponíveis, busca-se dividir o banco de dados em um conjunto de grupos ou classes. Com a associação da impressão digital a uma determinada classe, o banco de dados é reduzido ao conjunto de registros correspondentes a ela.

A classificação de impressões digitais consiste no problema de associar uma dada impressão a uma classe de maneira consistente e confiável. Em geral, sistemas de classificação para impressão digital são baseados em características globais da impressão. Diversas abordagens já foram propostas para classificação de impressões digitais: abordagens estruturais, sintáticas, estatísticas, baseadas em regras, em redes neurais, entre outras.

Neste contexto, este trabalho voltou-se para a implementação de um sistema estrutural de classificação, visando contribuir com o sistema de reconhecimento desenvolvido pela empresa VSoft Tecnologia, no que se refere a tempo de processamento e taxa de erro.

No Capítulo 2 são apresentados os objetivos associados ao estágio desenvolvido e aqui reportado. No Capítulo 3 é apresentada uma breve descrição da empresa. No Capítulo 4, são fundamentados conceitos considerados importantes para o sistema implementado. Em seguida, no Capítulo 5, as atividades realizadas são descritas, à medida em que são apresentados os resultados obtidos. Finalmente, o Capítulo 6 conclui este relatório.

## 2 Objetivo

O estágio realizado na empresa VSoft Tecnologia teve como principal objetivo a implementação do sistema estrutural de classificação desenvolvido por [Lumini, Maio e Maltoni, 1999](#). Trata-se de um sistema de classificação baseada no casamento inexato de grafos, que busca facilitar o processo de reconhecimento de impressões a partir da discriminação das respectivas classes.

Especificamente, podem ser destacadas as seguintes metas, correspondentes aos sub-sistemas proposto no artigo referenciado:

- Extrair imagem de orientação a partir de uma impressão digital;
- Segmentar a imagem de orientação em regiões segundo uma dada função de custo;
- Construir grafo conexo a partir da segmentação obtida;
- Definir banco de dados de teste;
- Criar grafos de referência (template) para cada classe tratada;
- Classificar grafo obtido dentro do conjunto de templates criados;



## 3 A Empresa

A Vsoft Tecnologia é uma empresa que atua no segmento de software, como fornecedora independente e especializada em identificação biométrica.

Sediada em João Pessoa - Paraíba, a Vsoft foi constituída em 2000 e se especializou, ao longo dos anos, no desenvolvimento de software voltado ao processamento digital de imagens, contando hoje com um corpo funcional de cerca de 40 funcionários.

A biometria tem sido seu foco principal, com a pesquisa e desenvolvimento de algoritmos de reconhecimento de faces e de impressões digitais. em parceria com a Universidade Federal da Paraíba (UFPB).



Figura 1 – Logo VSoft Tecnologia®

Em parceria com o Visio (Laboratório de Visão Computacional) da UFPB, a Vsoft vem trabalhando continuamente na pesquisa e desenvolvimento do produto de tecnologia de reconhecimento de faces e impressões digitais *BioPass*.

O *BioPass* é testado no FVC (*Fingerprint Verification Competition*), uma das mais importantes avaliações do mercado internacional em biometria, mantida pelo *Biometric System Laboratory* da Universidade de Bolonha (*Università di Bologna*), Itália. Também já teve um de seus módulos certificado pelo órgão americano FBI (*Federal Bureau of Investigation*).



## 4 Fundamentação

Este capítulo apresenta os principais conceitos e paradigmas envolvidos no sistema proposto, no intuito de prover maior embasamento teórico ao que será explanado posteriormente. Serão aqui descritos, em termos gerais, os sistemas biométricos, alguns dos erros neles envolvidos, a representação de uma impressão digital, a definição de sua imagem de orientação e o esquema de classificação Galton-Henry.

### 4.1 Sistemas biométricos

No desenvolvimento prático de um sistema biométrico, é importante determinar como o indivíduo será reconhecido. A depender do contexto da aplicação, o sistema é dito de *verificação* ou de *identificação*.

No primeiro tipo, o indivíduo alega sua identidade e esta é autenticada por meio de uma comparação entre as características biométricas capturadas e a referência biométrica previamente registrada no sistema. Tem-se, portanto, uma comparação um-para-um, que levará à aceitação ou rejeição da alegação feita. Um exemplo de um sistema de verificação é o reconhecimento em um caixa eletrônico.

Já em sistemas de identificação o reconhecimento se dá por meio da busca, ao longo de todo o banco de dados do sistema, por um registro que corresponda às características biométricas lidas. Trata-se de uma comparação um-para-muitos, que decidirá se um dado indivíduo está registrado ou não e, em caso afirmativo, retornará o identificador do registro correspondente. Um exemplo é o controle de acesso dentro de uma empresa.

### 4.2 Erros em sistemas biométricos

Há três razões principais que explicam os erros observados em um sistema biométrico: limitação de informação – possivelmente devido à capacidade intrínseca do sinal do identificador biométrico; limitação na representação – o esquema ideal de representação deveria reter toda a invariância e informação discriminatória das medições feitas; e limitação no modelo da invariância – o casamento ideal modelaria perfeitamente a relação de invariância entre os diferentes padrões de uma mesma classe (usuário).

Erros podem ocorrer nos diferentes módulos de um sistema biométrico, como na captura, na extração de características, na criação de templates e no casamento. Nesta seção, serão apresentados os erros de casamento e verificação, visto que são os mais relevantes para as atividades desenvolvidas.

### 4.2.1 Erros no módulo de casamento

O resultado de um casamento é tipicamente uma pontuação (*score*) no intervalo  $[0, 1]$  que quantifica a similaridade entre o conjunto de características lidas e o template registrado. Quanto mais próximo de 1, maior a certeza de que as características vêm do mesmo dedo. A decisão é regulada pela escolha de um limiar (*threshold*)  $\tau$ : pares com scores iguais ou superiores a  $\tau$  são denominados casados (*matching pairs*), pertencendo ao mesmo dedo, enquanto que aqueles com scores inferiores a  $\tau$  são chamados não-casados (*non-matching pairs*), isto é, pertencem a dedos diferentes.

O módulo de casamento, operando no modo de comparação um-para-um, pode acarretar em dois tipos de erros: (i) decidir que as características de dois dedos diferentes correspondem ao mesmo dedo (falso positivo) e (ii) decidir que as características de dois conjuntos referentes ao mesmo dedo correspondem a dedos distintos (falso negativo).

Quando o sistema opera no modo de identificação, o casamento pressupõe uma comparação um-para-muitos, que em sua forma mais simples, pode ser vista como um conjunto de comparações um-para-um. Se o módulo de casamento, ao receber um conjunto de características (*features*) de um dedo  $A$  e um conjunto de templates que inclui pelo menos um registro de  $A$ , retornar uma lista de candidatos vazia, trata-se de um erro de identificação do tipo falso negativo. Por outro lado, se o conjunto de templates não incluir registros de  $A$  e, mesmo assim, o módulo de casamento retornar uma lista não vazia de candidatos, tem-se um erro de identificação do tipo falso positivo.

### 4.2.2 Taxas de erro de verificação

Os conceitos de erro falso positivo e falso negativo foram descritos em termos de uma comparação um-para-um, típico de um processo de verificação. Tais erros, portanto, também podem ser considerados erros de verificação. Seja  $\mathbf{T}$  o registro (*template*) biométrico de um indivíduo e  $\mathbf{I}$  conjunto de características de verificação. Definem-se as hipóteses nula e alternativa:

$H_0$ :  $\mathbf{I} \neq \mathbf{T}$ , conjunto de verificação e conjunto template não são do mesmo dedo;

$H_1$ :  $\mathbf{I} = \mathbf{T}$ , conjunto de verificação e conjunto template são do mesmo dedo;

As decisões associadas são:  $D_0$  (não casado) e  $D_1$  (casado). Se o score for alto o suficiente (superior ao threshold  $\tau$ ), decide-se por  $D_1$ ; caso contrário, por  $D_0$ . Esta formulação inerentemente leva a dois tipos de erros: o tipo I – falso positivo (decidir por  $D_1$  quando  $H_0$  é verdadeiro) e o tipo II – falso negativo (decidir por  $D_0$  quando  $H_1$  é verdadeiro).

A taxa de falsos positivos (*False Match Rate* – *FMR*) é dada pela probabilidade do erro de tipo I (também chamada de nível de significância da hipótese de teste) e a taxa de falsos negativos (*False Non-Match Rate* – *FNMR*) pela probabilidade do erro de tipo II:

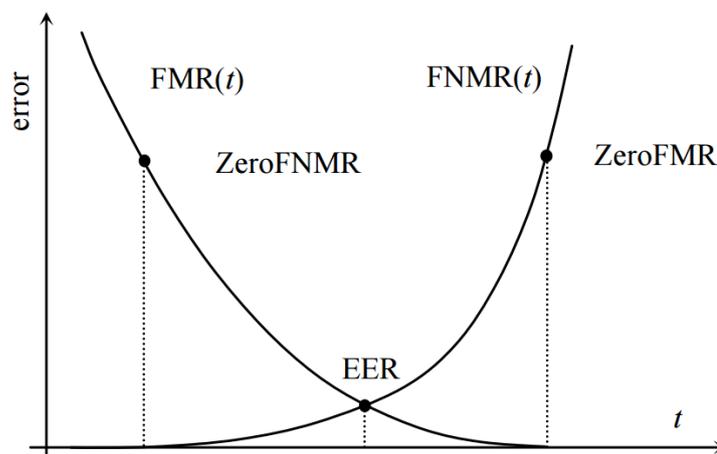
$$\begin{aligned} FMR &= P(D_1|H_0) \\ FNMR &= P(D_0|H_1). \end{aligned} \quad (4.1)$$

Vale destacar que  $(1 - FNMR)$  é também chamado de potência do teste. Ambas as taxas são funções do limiar  $\tau$  do sistema, de modo que deveriam ser referidas, na verdade, como  $FMR_\tau$  e  $FNMR_\tau$ . Se o limiar é reduzido, no intuito de oferecer maior tolerância a ruído e a variações de entrada, a  $FMR_\tau$  aumenta. Por outro lado, se  $\tau$  é aumentado, para tornar o sistema mais seguro, a  $FNMR_\tau$  cresce.

Alguns índices são utilizados para medir a acurácia do sistema de verificação:

- *Taxa de erro igual (Equal-Error Rate - EER)*: é a taxa de erro para o limiar  $\tau$  onde as taxas de falso positivo e falso negativo coincidem:  $FMR_\tau = FNMR_\tau$  (ver Figura 1.6).
- *ZeroFNMR*: é definido como o menor valor  $FMR$  para o qual não ocorrem falsos negativos, isto é,  $FNMR = 0$ .
- *ZeroFMR*: é definido como o menor valor  $FNMR$  para o qual não ocorrem falsos positivos, isto é,  $FMR = 0$ .

A Figura 2 exemplifica as curvas  $FMR(\tau)$  e  $FNMR(\tau)$ , ilustrando os comportamentos distintos que cada um delas tem em relação ao limiar  $\tau$ . Nela, são também marcados os pontos referentes às taxas  $EER$ ,  $ZeroFNMR$  e  $ZeroFMR$ .

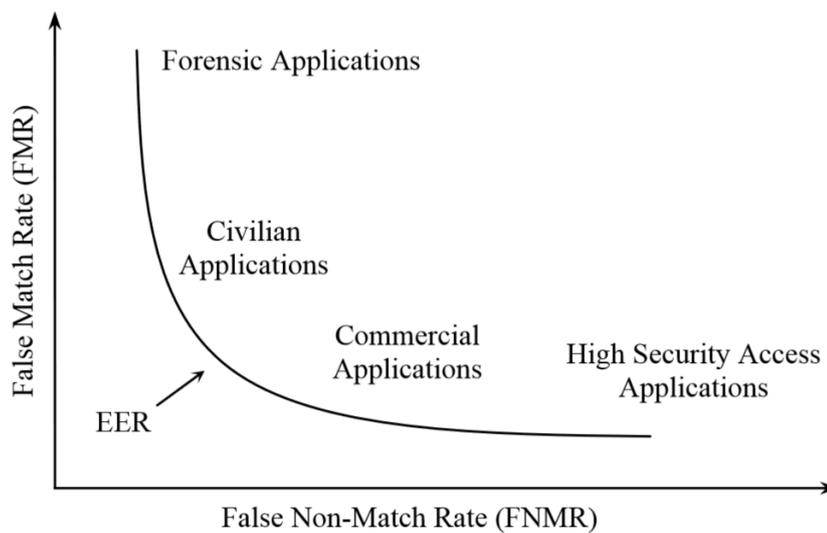


Fonte: [Maltoni et al., 2009](#)

Figura 2 – Um exemplo de curvas  $FMR(t)$  e  $FNMR(t)$ , onde são marcados os pontos de  $EER$ ,  $ZeroFNMR$  e  $ZeroFMR$ .

Os requisitos impostos sobre os valores de  $FMR$  e  $FNMR$ , priorizando um em detrimento do outro, dependem da aplicação. Em identificação criminal, por exemplo,

a taxa de falsos negativos (*FNMR*) é a mais preocupante: busca-se evitar ao máximo que um criminoso passe despercebido, mesmo que isso leve a um extenso conjunto de falsos positivos identificados pelo sistema. Em uma aplicação de controle de acesso de alta segurança, por outro lado, o principal objetivo é evitar impostores, reduzindo a taxa de falsos positivos. Há ainda casos intermediários, onde ambas as taxas são relevantes, como na verificação de um indivíduo em um caixa eletrônico: um falso positivo pode levar à perda de centenas de reais enquanto que uma alta taxa de falsos negativos pode gerar desconforto para o cliente genuíno. A Figura 3 ilustra o balanço entre as taxas *FMR* e *FNMR* para diferentes aplicações.



Fonte: [Maltoni et al., 2009](#)

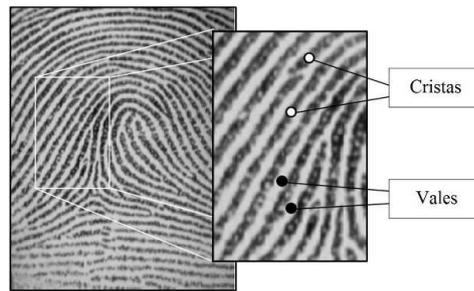
Figura 3 – Pontos típicos de operação para diferentes aplicações.

### 4.3 Representação de uma impressão digital

Uma impressão digital é a reprodução da aparência exterior da epiderme da ponta do dedo. A característica estrutural mais evidente de uma impressão é o padrão intercalado de cristas (*ridges*) e vales (*valleys*). Na imagem de uma impressão digital, cristas são escuras enquanto que vales são claros, como mostrado na Figura 4.

A representação é a essência do design de um sistema de reconhecimento de impressões digitais. Busca-se determinar um espaço de medições/características (*features*) no qual imagens de impressões digitais de um mesmo dedo formem um compacto agrupamento (*cluster*), apresentando baixa variação intra-classe, e imagens de dedos diferentes ocupem diferentes porções do espaço, com alta variação inter-classe.

O padrão de uma impressão digital, quando analisado em escalas diferentes, apresenta três níveis distintos de características:



Fonte: Adaptado de [Maltoni et al., 2009](#)

Figura 4 – Cristas e vales em uma impressão digital.

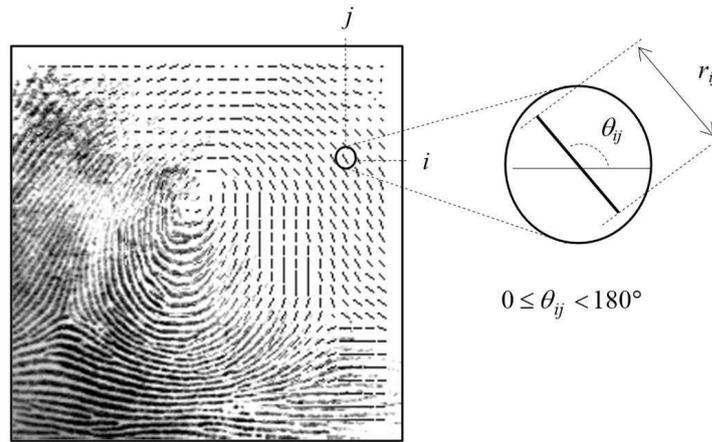
1. *Nível 1*: em um nível global, as cristas normalmente seguem em paralelo, mas exibem uma ou mais regiões onde assumem formas distintas, caracterizados por alta curvatura, terminações frequentes, etc. Estas regiões, chamadas de regiões singulares ou, simplesmente, singularidades, podem ser classificadas em três topologias: *loop*, *delta* e *whorl*. Formato da impressão digital, imagem de orientação e imagem de frequência também são características de nível global.
2. *Nível 2*: em um nível local, foram identificadas dezenas de características denominadas *minúcias*. Os dois tipos mais proeminentes de minúcias são as terminações e as bifurcações. A terminação de uma crista é definida como o ponto onde a crista bruscamente termina, enquanto que a bifurcação de uma crista consiste no ponto onde ela se subdivide em cristas ramificadas.
3. *Nível 3*: em um nível bastante refinado, detalhes intra-cristas podem ser detectados. Um dos detalhes mais importantes neste nível são os poros de suor, dos quais o formato e a posição são considerados bastante distintivos.

#### 4.3.1 Orientação local de cristas e a Imagem de Orientação

A orientação local de uma crista no pixel  $[x, y]$  é o ângulo  $\theta_{xy}$  que as cristas formam com o eixo horizontal, dentro de uma pequena vizinhança arbitrária centrada em  $[x, y]$ . Como impressões são naturalmente não direcionadas,  $\theta_{xy}$  é uma direção não orientada no intervalo  $[0, 180^\circ)$ .

Em vez de calcular orientação de cristas em cada pixel, a estimativa é feita em posições discretas, como em blocos. A imagem de orientação de uma impressão digital é uma matriz  $\mathbf{D}$  cujos elementos codificam a orientação local das cristas da impressão. Cada elemento  $\theta_{ij}$  correspondente ao nó  $[i, j]$  de uma malha retangular localizado sobre o pixel  $[x_i, y_j]$  consiste na orientação média das cristas na vizinhança de  $[x_i, y_j]$  (ver Figura 5).

Muitas vezes, um valor adicional  $r_{ij}$  é associado a cada elemento para exprimir a confiança (ou consistência) da orientação. O valor é baixo para regiões ruidosas e corrompidas e alto para regiões de boa qualidade.

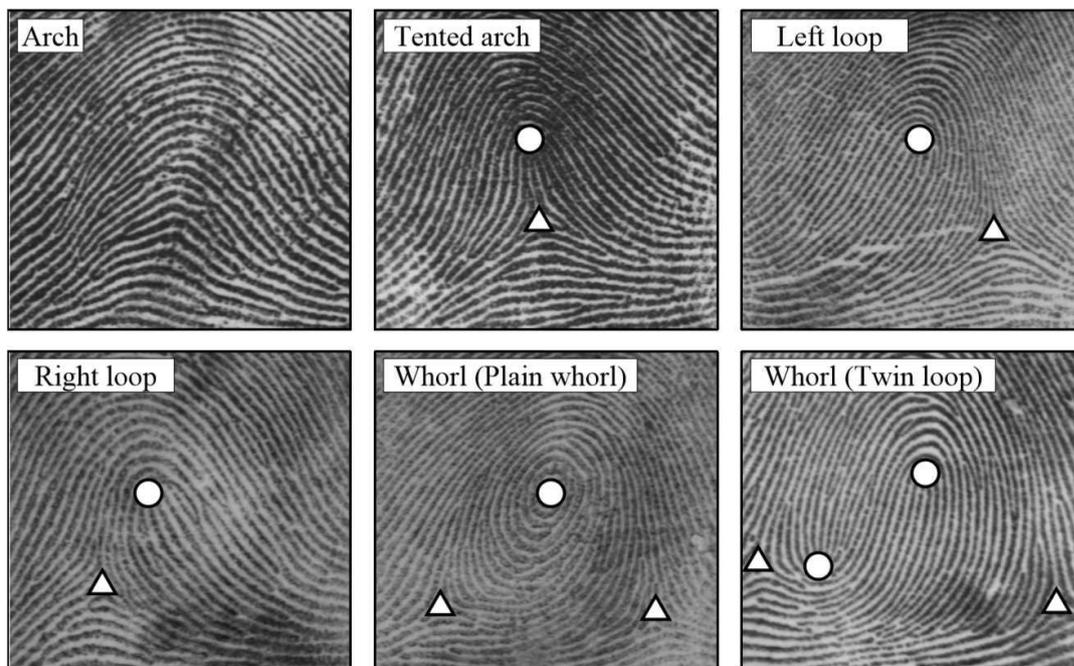


Fonte: [Maltoni et al., 2009](#)

Figura 5 – Uma imagem de impressão combinada com sua correspondente imagem de orientação, calculada sobre uma malha de tamanho 16x16.

#### 4.4 Esquema de Classificação Galton-Henry

O esquema de classificação Galton-Henry ([Maltoni et al., 2009](#)) foi adotado em vários países: na verdade, a maioria dos esquemas de classificação atualmente usados por agências pelo mundo são variantes do esquema Galton-Henry. A Figura 6 mostra as cinco classes mais comuns do esquema de classificação Galton-Henry: arco (*arch*), arco angular (*tented arch*), presilha esquerda (*left loop*), presilha direita (*right loop*) e verticilo (*whorl*).



Fonte: [Maltoni et al., 2009](#)

Figura 6 – As cinco classes de impressões digitais comumente utilizadas. Dois tipos de impressões whorl são mostradas (plain e twin).

- Uma impressão do tipo arco possui cristas que entram por um lado, se elevam suavemente, e saem do lado oposto ao que entraram. Arcos não apresentam singularidades, como loops ou deltas.
- Uma impressão do tipo arco angular é similar ao arco (plano), com exceção de que pelo menos uma crista apresentará elevada curvatura e de que haverá um *loop* e um delta.
- Uma impressão do tipo presilha (*loop*) tem uma ou mais cristas que entram por um lado, se curvam e saem pelo mesmo lado que entraram. Estão presentes um *loop* e um delta. Assume-se que o delta está a sul do *loop*.
- Uma impressão do tipo verticilo (*whorl*) contém pelo menos uma crista que apresenta um caminho fechado de  $360^\circ$  ao redor do centro da impressão. Dois *loops* e dois deltas podem ser encontrados.



## 5 Atividades realizadas

Este capítulo relata as atividades realizadas ao longo do estágio. Como citado anteriormente, o trabalho teve como principal objetivo a implementação do sistema estrutural de classificação desenvolvido por [Lumini, Maio e Maltoni, 1999](#). Ao que segue, as atividades correspondem aos subsistemas propostos no trabalho citado.

### 5.1 Visão geral do trabalho implementado

O trabalho implementado durante o estágio consiste em um sistema de classificação baseada no casamento inexato de grafos, que busca facilitar o processo de reconhecimento de impressões a partir da discriminação das respectivas classes, reduzindo o espaço efetivo de busca.

O método gera, a partir da imagem de orientação, uma versão mais compacta dela, agrupando elementos similares em regiões “homogêneas”. Com esse agrupamento, a impressão digital é segmentada em regiões que, por sua vez, serão utilizadas para construção do grafo que a representa.

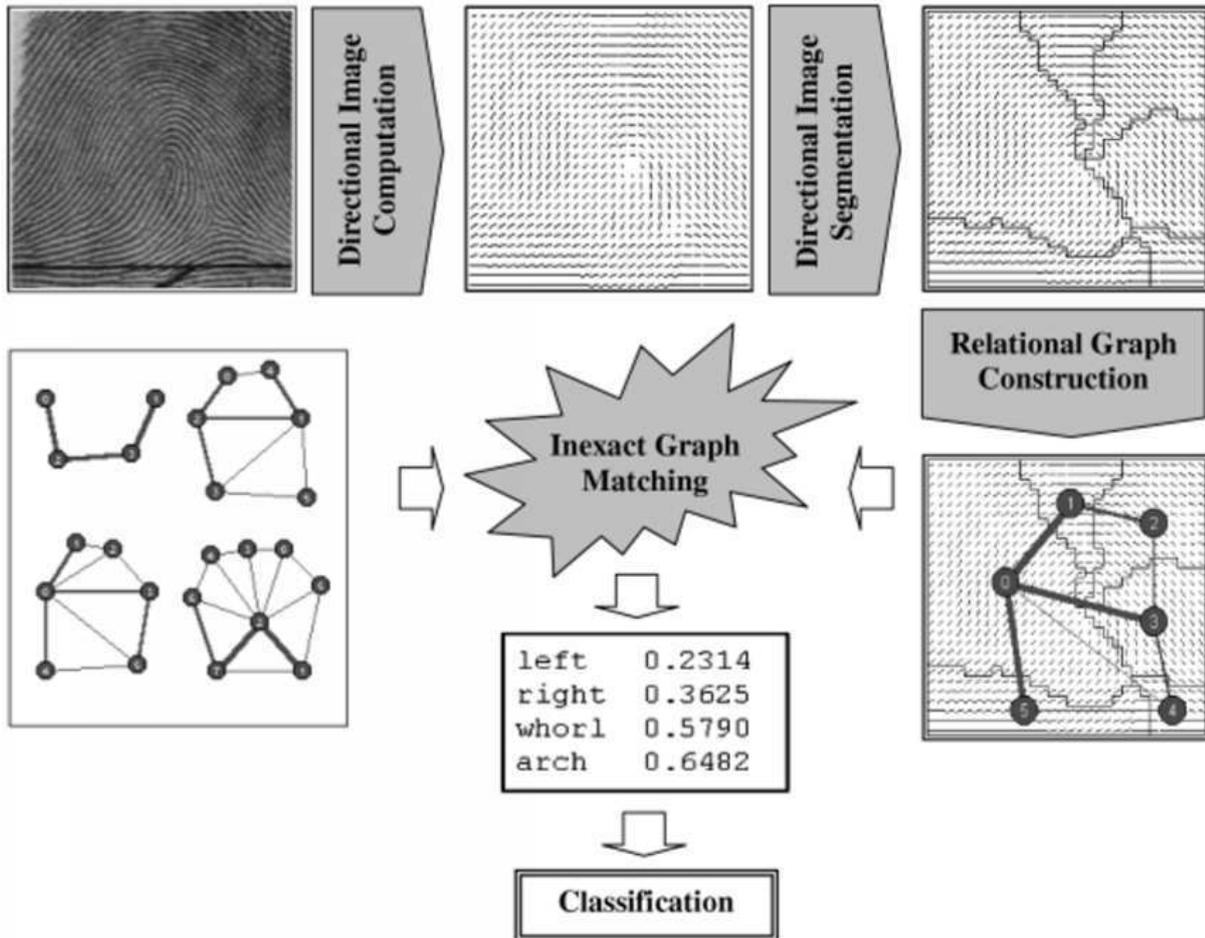
A Figura 7 apresenta os principais passos na abordagem estrutural implementada. Primeiramente, calcula-se a imagem de orientação a partir da impressão digital que se deseja classificar. Em seguida, a imagem de orientação é segmentada em regiões, de acordo com uma determinada função de custo. Da segmentação, constrói-se um grafo conexo, associando um nó a cada região e um arco a cada par de regiões.

Em seguida, a partir do casamento inexato de grafos, estima-se a “distância” entre o grafo obtido e o grafo de referência para cada uma das classes. Os grafos de referência são previamente prototipados a partir de um banco de impressões representativas de cada classe.

### 5.2 Obtenção da imagem de orientação

A maneira mais natural de extrair a orientação local das cristas é baseada no cálculo dos gradientes da imagem de impressão digital. O gradiente  $\nabla(x, y)$  no ponto  $[x, y]$  da imagem  $\mathbf{I}$  é um vetor bidimensional  $[\nabla_x(x, y), \nabla_y(x, y)]$ , onde as componentes  $\nabla_x$  e  $\nabla_y$  correspondem às derivadas de  $\mathbf{I}$  em  $[x, y]$  com respeito às direções  $x$  e  $y$ , respectivamente.

Como o ângulo de fase associado ao gradiente denota a direção de máxima variação de intensidade, a direção  $\theta$  de uma borda que cruza uma região centrada em  $[x, y]$  será ortogonal ao ângulo de fase do gradiente no mesmo ponto. A estimativa mais direta para  $\theta$



Fonte: [Maltoni et al., 2009](#)

Figura 7 – Os passos principais na abordagem estrutural trabalhada. Resultados exemplificam a classificação de uma impressão do tipo *left loop*.

é pelo cálculo do arcotangente de  $\nabla_y/\nabla_x$ . Porém, este procedimento traz dois problemas: a não-linearidade da função utilizada e a descontinuidade em torno de  $90^\circ$ .

Além disso, estimar as orientações individualmente levam a uma representação muito refinada do padrão de cristas e vales, geralmente muito sensível a ruído. Por outro lado, tomar a média de tais estimativas não seria significativo devido à circularidade dos ângulos: a orientação média entre  $5^\circ$  e  $175^\circ$  não é  $90^\circ$ , como uma média aritmética sugeriria, mas  $0^\circ$ . Nem sempre o conceito de média de ângulos é bem definida: qual seria, por exemplo, a média entre  $0^\circ$  e  $90^\circ$ :  $45^\circ$  ou  $135^\circ$ ?

Para realizar a média entre estimativas de gradientes, a ideia básica utilizada foi a de dobrar os ângulos, de modo que cada estimativa de orientação fosse codificada por um vetor do tipo:

$$\mathbf{d} = [r \cdot \cos(2\theta), r \cdot \sin(2\theta)] \quad (5.1)$$

Dessa maneira, usando  $2\theta$  no lugar de  $\theta$ , a circularidade dos ângulos é compensada. O fator  $r$  pode ser utilizado para denotar a força ou confiança da estimativa, sendo, por exemplo, a norma quadrática do gradiente:  $\nabla_x^2 + \nabla_y^2$ .

Codificando as orientações deste modo, pode-se tomar a média de ângulos dentro de uma janela  $W$  de dimensões  $n \times n$  para obter uma estimativa  $\bar{\mathbf{d}}$  tratando cada componente separadamente:

$$\bar{\mathbf{d}} = \left[ \frac{1}{n^2} \sum_W r \cdot \cos(2\theta), \frac{1}{n^2} \sum_W r \cdot \sin(2\theta) \right] \quad (5.2)$$

Já a obtenção de  $\theta_{ij}$  é feita pelo cálculo da orientação de crista dominante na janela  $W$  de dimensões  $17 \times 17$  centrada em  $[x_i, y_j]$ , por meio da combinação de múltiplas estimativas de gradiente, conforme descrito a seguir:

$$\begin{aligned} \theta_{ij} &= 90^\circ + \frac{1}{2} \cdot \text{atan2}(2G_{xy}, G_{xx} - G_{yy}), \\ G_{xy} &= \sum_{h=-8}^8 \sum_{k=-8}^8 \nabla_x(x_i + h, y_j + k) \cdot \nabla_y(x_i + h, y_j + k), \\ G_{xx} &= \sum_{h=-8}^8 \sum_{k=-8}^8 \nabla_x(x_i + h, y_j + k)^2, \\ G_{yy} &= \sum_{h=-8}^8 \sum_{k=-8}^8 \nabla_y(x_i + h, y_j + k)^2 \end{aligned} \quad (5.3)$$

Na Equação 5.3,  $\nabla_x$  e  $\nabla_y$  são as componentes do gradiente nas direções  $x$  e  $y$ , respectivamente, que podem ser calculadas, por exemplo, a partir de máscaras Sobel  $3 \times 3$ . Seguindo uma recomendação da *OpenCV*, no entanto, foram utilizadas máscaras Schaar de mesmas dimensões (Itseez, ). A função Scharr, tão rápida quanto a Sobel e com maior acurácia, é implementada a partir dos seguintes núcleos de convolução:

$$G_x = \begin{bmatrix} -3 & 0 & +3 \\ -10 & 0 & +10 \\ -3 & 0 & +3 \end{bmatrix}, \quad G_y = \begin{bmatrix} -3 & -10 & -3 \\ 0 & 0 & 0 \\ +3 & +10 & +3 \end{bmatrix} \quad (5.4)$$

Na Figura 8 a obtenção da imagem de orientação é exemplificada para duas impressões (um arco e uma presilha esquerda – *left loop*). Estas, assim como as demais utilizadas neste trabalho, foram geradas automaticamente pelo software *Anguli* (Namdeo, 2012). Nas três primeiras colunas estão as impressões originais (a) e as componentes horizontal (b) e vertical (c) de seus gradientes.

Com as componentes do gradiente, as expressões para  $G_{xy}$ ,  $G_{xx}$  e  $G_{yy}$  na Equação 5.3 podem ser calculadas e a orientação  $\theta_{ij}$  estimada para cada posição  $(i, j)$ . Com isso, é construída a imagem de orientação, apresentada na Figura 8 (d), que associa a cada pixel um ângulo no intervalo  $[0^\circ, 180^\circ]$ , mapeado na escala de cinza  $[0, 255]$ . Portanto, há na imagem de orientação um contraste visual entre os extremos da escala de cinza ( $0^\circ$  e  $180^\circ$ ).

Vale enfatizar que a imagem de orientação oferece uma precisão no nível de pixel, isto é, tem-se uma estimativa de orientação para cada pixel. Neste trabalho, no entanto, não é interessante fazer uso deste nível de precisão. Assim, a partir desta imagem de

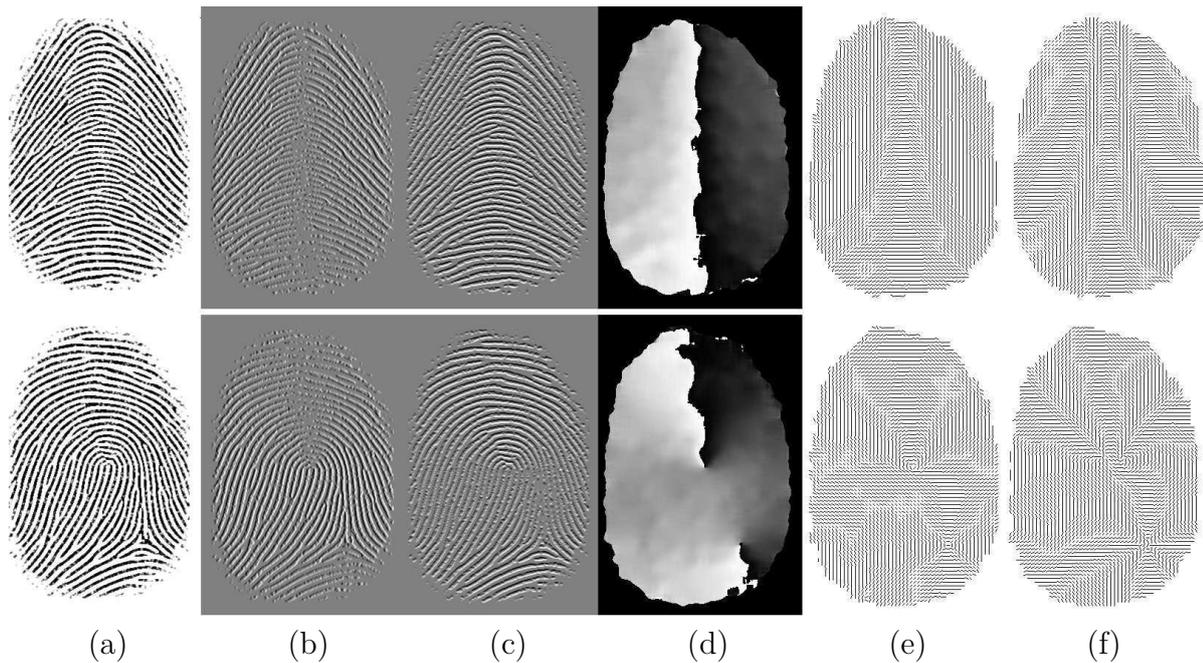


Figura 8 – Obtenção da imagem de orientação para duas impressões digitais: na primeira linha, uma impressão do tipo arco; na segunda, uma do tipo presilha esquerda (left loop). Ao longo de cada linha, tem-se: (a) imagem original, imagens gradiente no eixo  $x$  (b) e  $y$  (c), imagem de orientação por pixel (d), imagem de orientação por bloco (e) e imagem de orientação por bloco suavizada (f).

orientação, pode-se gerar uma imagem no nível de blocos: define-se um bloco de, por exemplo,  $5 \times 5$  pixels e, a cada bloco, associa-se a média das orientações de seus elementos. Na Figura 8 (e) são apresentadas as respectivas imagens de orientação a nível de bloco ( $5 \times 5$ ).

Finalmente, decidiu-se por aplicar à imagem de orientação uma simples suavização a nível de blocos. Utilizando uma janela de  $7 \times 7$  blocos, a orientação representativa de cada bloco é atualizada pela orientação média da janela. O resultado desta suavização é apresentado na Figura 8 (f). Com isso, obtém-se a imagem de orientação (por bloco) de uma impressão digital.

### 5.3 Segmentação da imagem de orientação

O objetivo da segmentação é particionar a imagem de orientação em regiões homogêneas, isto é, regiões de formatos regulares contendo elementos com orientações similares. Dada uma imagem de orientação  $D$ , uma segmentação de  $D$  é definida como um particionamento  $R = \{R_1, R_2, \dots, R_n\}$  de  $D$  em  $n$  regiões conexas  $R_i$ . Uma região  $R_i$  é dita conexa se, para cada par  $(d_i, d_j)$  de elementos em  $R_i$ , existe um caminho de vizinhança-4 que conecta  $d_i$  a  $d_j$ .

O custo total  $C_{tot}(R)$  de uma segmentação é a soma de custos  $C(R_i)$  de cada região:

$$\begin{aligned} C_{tot}(R) &= \sum_{i=1}^n C(R_i), \\ C(R_i) &= C_0 + w_d \cdot C_d(R_i) + w_r \cdot C_r(R_i) \end{aligned} \quad (5.5)$$

onde  $C_d(R_i)$  e  $C_r(R_i)$  quantificam, respectivamente, a não-homogeneidade na orientação dos elementos que pertencem a  $R_i$  e a regularidade de formato da região, sendo  $w_d$  e  $w_r$  seus respectivos pesos e  $C_0$  um limiar (threshold) correspondente ao custo de uma região de um único elemento. O valor de  $C_0$  impõe uma penalidade proporcional ao número de regiões em  $R$ , evitando que a segmentação se dê em muitas regiões pequenas.

Os custos  $C_d(R_i)$  e  $C_r(R_i)$  são definidos a seguir:

$$C_d(R_i) = Var(R_i) \quad (5.6)$$

$$C_r(R_i) = \frac{Perímetro(R_i)}{\sqrt{4\pi \cdot Área(R_i)}} \quad (5.7)$$

em que  $Var(R_i)$  relaciona-se com a variância na orientação dos elementos de  $R_i$ ,  $Área(R_i)$  e  $Perímetro(R_i)$  denotando, respectivamente, a área e o perímetro da região  $R_i$ . Enquanto  $C_d(R_i)$  é minimizado por regiões com elementos de mesma orientação,  $C_r(R_i)$  é minimizado por regiões circulares:  $C_r(R_i) = 1$  para um círculo e cresce para formas mais alongadas. O cálculo da variância é feito conforme descrito no Apêndice A de [Lumini, Maio e Maltoni, 1999](#). Basicamente, a variância de uma região  $R_i$  contendo  $n_i$  elementos direcionais  $\mathbf{v} = [v_x, v_y]$  é definida como:

$$Var(R_i) = \sum_{\mathbf{v} \in R_i} \left\{ \left[ \left( \frac{v_x}{|\mathbf{v}|} - \frac{m_x}{|\mathbf{m}|} \right)^2 + \left( \frac{v_y}{|\mathbf{v}|} - \frac{m_y}{|\mathbf{m}|} \right)^2 \right] |\mathbf{v}|^2 \right\} \quad (5.8)$$

onde

$$m_x = \frac{1}{n_i} \sum_{\mathbf{v} \in R_i} v_x, \quad m_y = \frac{1}{n_i} \sum_{\mathbf{v} \in R_i} v_y, \quad \mathbf{m} = [m_x, m_y] \quad (5.9)$$

De todas as possíveis segmentações que podem ser feitas, denota-se a *segmentação ótima* como aquela que minimiza o custo total  $C_{tot}(R)$ . A Figura 9 apresenta o pseudo-código do algoritmo de segmentação e do procedimento de reatribuição de elementos, descritos a seguir.

A segmentação passa por uma varredura em espiral dos elementos da imagem de orientação, de dentro para fora. Caso conhecido, pode-se adotar o *core* da impressão digital como elemento inicial da varredura; do contrário, toma-se o elemento central da imagem. Com o elemento inicial ( $d_1$ ) é criada a primeira região da segmentação ( $R_1$ ). Neste momento, portanto, tem-se apenas uma região com um único elemento e custo  $C_0$ .

**SEGMENTATION**

```

{ d1:=spiral scan starting element;
  R1:= { d1 };
  while (the spiral scan is not completely performed)
    { dcurr:=get next spiral scan element;
      ΔCRbest := minRi ∈ Radj(dcurr) { C(Ri + {dcurr}) - C(Ri) };
      if ( ΔCRbest > C0 ) { Rnew := { dcurr } };
      else { Rbest := Rbest + { dcurr } };
      reassign (dcurr);
    }
  repeat k times
    { d1:=TV scan starting element;
      while (TV scan is not completely performed)
        { dcurr:=get next TV scan element;
          if (dcurr is a border element) reassign (dcurr);
        }
    }
}

```

**reassign (d<sub>i</sub>)**

```

{ count:=0;
  L:=Eadj(di);
  while ( ( L ≠ 0 ) ∧ ( count < max_reassignments ) )
    { dcurr:=head(L);
      if (dcurr can be reassigned )
        { Rcurr:=region containing (dcurr);
          ΔCRcurr := C( Rcurr ) - C( Rcurr - {dcurr} );
          ΔCRbest := minRi ∈ Radj(dcurr) { C(Ri + {dcurr}) - C(Ri) };
          if ( ΔCRbest < ΔCRcurr )
            { Rbest := Rbest + { dcurr };
              Rcurr := Rcurr - { dcurr };
              L := L ∪ Eadj(di);
              count := count + 1;
            }
        }
      remove dcurr from L;
    }
}

```

Fonte: [Maio e Maltoni, 1996](#)

Figura 9 – Pseudo-código do algoritmo de segmentação e do procedimento de reatribuição.

A partir do elemento inicial, seguindo o padrão espiral, avalia-se, para cada elemento  $d_{curr}$ , qual opção é menos custosa: incorporá-lo a uma das regiões vizinhas já existentes (naquela em que o incremento de custo seja mínimo) ou criar uma nova região para ele, incrementando o custo em  $C_0$ . Basicamente, uma nova região deverá ser criada se o custo  $C_0$  associada à criação for inferior ao incremento mínimo de custo associado à incorporação do elemento a uma região vizinha. Vale ressaltar que, no intuito de garantir a criação de regiões conexas, apenas regiões vizinhas são consideradas na incorporação de um novo elemento:  $R_{adj}(d_{curr})$  retorna as regiões que estão na vizinhança-4 do elemento  $d_{curr}$ .

Após cada atribuição, o algoritmo pode ainda rearranjar sua vizinhança-8 para reduzir o custo. O procedimento de reatribuição ( $reassign(d_i)$ ) constrói, para um dado elemento  $d_i$ , uma lista  $L$  contendo os elementos que estão na vizinhança-8 de  $d_i$  e que se encontram na borda de duas ou mais regiões ( $L = E_{adj}(d_i)$ ).

Então, avalia-se se cada elemento  $d_{curr}$  de  $L$  pode ser reatribuído, isto é, se sua reatribuição não resulta na partição da região a que pertence. Caso a reatribuição seja válida, deve-se comparar o decremento de custo relativo à remoção de  $d_{curr}$  de sua região atual com o (mínimo) incremento de custo relativo à sua inserção em uma região vizinha ( $R_{adj}$ ). Caso se decida pela reatribuição de  $d_{curr}$  a outra região, devem ser acrescentados à cauda de  $L$  os elementos retornados por  $E_{adj}(d_i)$  que já não existam em  $L$  ( $L = L \cup E_{adj}(d_i)$ ). Trata-se, portanto, de um procedimento recursivo. O parâmetro `max_reassignments` é utilizado para interromper o procedimento caso este não se estabilize, podendo se evitar sequências cíclicas de reatribuição.

Ao fim da varredura em espiral, todos os elementos estão atribuídos e tem-se uma segmentação consistente, isto é, com regiões conexas. Duas varreduras posteriores simples (no padrão de “varredura de TV”) são realizadas, avaliando a reatribuição dos elementos de borda, para aprimoramento da solução. Por fim, se a solução contém várias regiões pequenas (provavelmente oriundas de ruído), estas podem ser integradas a regiões vizinhas.

Na Figura 10 são apresentados alguns exemplos de segmentações obtidas com este algoritmo. Tem-se a segmentação de uma impressão do tipo arco (i), com quatro regiões definidas, presilha esquerda (ii), com seis, e presilha direita (iii), com sete regiões. Nestas segmentações, os parâmetros utilizados foram  $C_0 = 50$ ,  $w_d = 30$  e  $w_r = 5$ .

## 5.4 Construção do grafo conexo

A partir do resultado obtido com a segmentação da imagem de orientação, informações consideradas relevantes a respeito das regiões encontradas, bem como das relações entre elas, podem ser estruturadas em um grafo relacional (conexo).

A segmentação da imagem de orientação representa uma base ótima para a descrição das impressões digitais a partir de tais grafos. Dada uma segmentação  $R = \{R_1, R_2, \dots, R_n\}$  define-se como grafo relacional de  $R$  a 4-tupla  $G = (V, A, f_V, f_A)$ , onde:

- $V = \{v_i | i = 1, \dots, n\}$  é um conjunto de nós, cada um correspondente a uma diferente região  $R_i$  de  $R$ :  $v_i \equiv R_i$ ;
- $A = \{a_{ij} | 1 \leq i < n, i < j \leq n, AdjP(R_i, R_j) > 0\}$  é um conjunto de arcos não direcionados; cada arco codifica uma relação de adjacência entre um par de regiões de  $R$ ; a função  $AdjP(R_i, R_j)$  retorna o comprimento (em blocos) do perímetro de adjacência (fronteira) entre as regiões  $R_i$  e  $R_j$ .

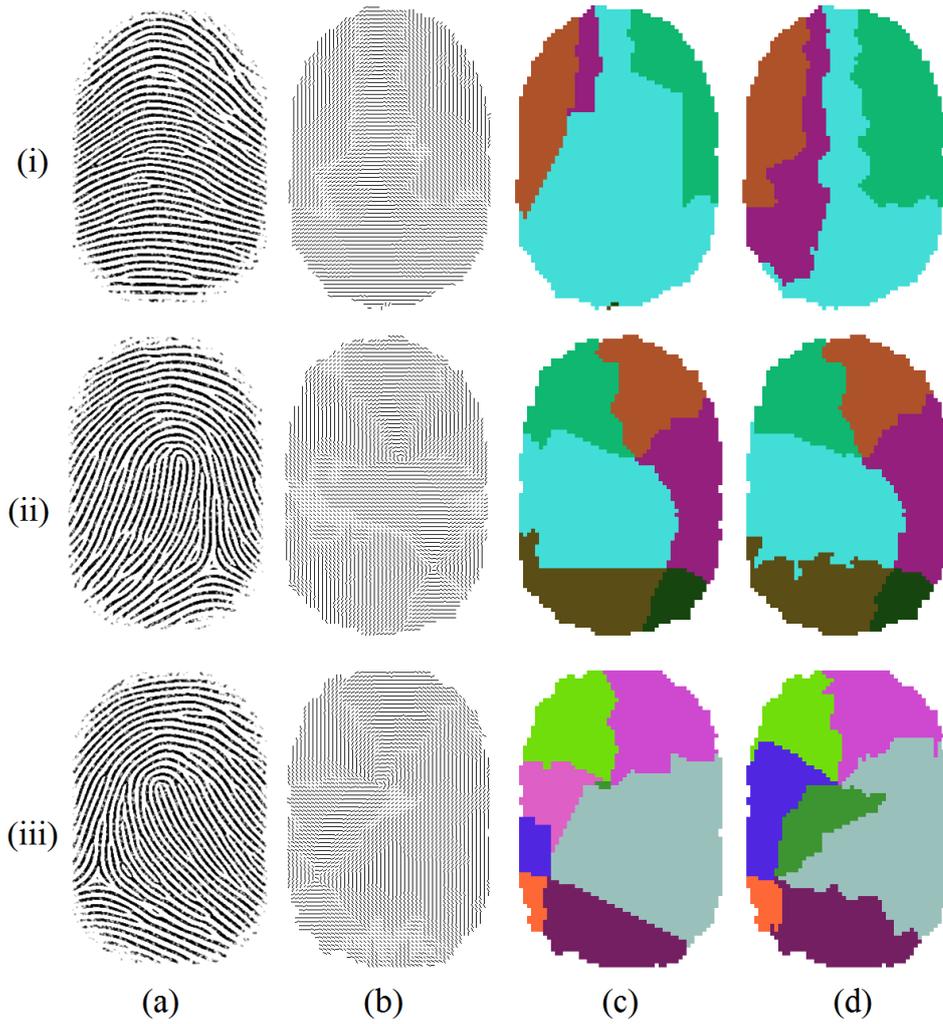


Figura 10 – Exemplo de segmentação para uma impressão do tipo (i) arco, (ii) presilha esquerda e (iii) presilha direita: (a) impressão original, (b) imagem de orientação, (c) resultado da segmentação espiral, (d) resultado final após varredura de TV e remoção de regiões pequenas. Custos totais das segmentações foram, respectivamente: 6135, 15843 e 16936.

- $f_V : V \rightarrow \mathbb{R}$  é uma função que rotula os nós em  $V$  como segue:

$$f_V(v_i) = [\text{Área}(R_i)] \quad (5.10)$$

- $f_A : A \rightarrow \mathbb{N} \times [-90^\circ, 90^\circ] \times \mathbb{N}$  é uma função que rotula os arcos em  $A$  como segue:

$$f_A(a_{ij}) = [\text{AdjP}(R_i, R_j), \Delta\text{dir}(R_i, R_j), \text{Bdist}(R_i, R_j)] \quad (5.11)$$

onde a função  $\text{Bdist}(R_i, R_j)$  retorna a distância Euclidiana entre os baricentros das regiões  $R_i$  and  $R_j$  e a função  $\Delta\text{dir}(R_i, R_j)$  retorna a diferença de fase entre as direções médias

$\theta_i, \theta_j \in [-90^\circ, 90^\circ[$ , calculada da seguinte forma:

$$\Delta_{dir}(R_i, R_j) = \begin{cases} \theta_i - \theta_j & \text{se } -90^\circ \leq \theta_i - \theta_j < 90^\circ \\ \theta_i - \theta_j + 180^\circ & \text{se } \theta_i - \theta_j < -90^\circ \\ \theta_i - \theta_j - 180^\circ & \text{se } \theta_i - \theta_j \geq 90^\circ \end{cases} \quad (5.12)$$

A Figura 11 apresenta uma visualização dos grafos gerados a partir das segmentações apresentadas na Figura 10. Neles, os círculos que representam os nós têm diâmetro proporcional à área da região correspondente, enquanto que as espessuras dos arcos são proporcionais ao perímetro de adjacência entre suas regiões.

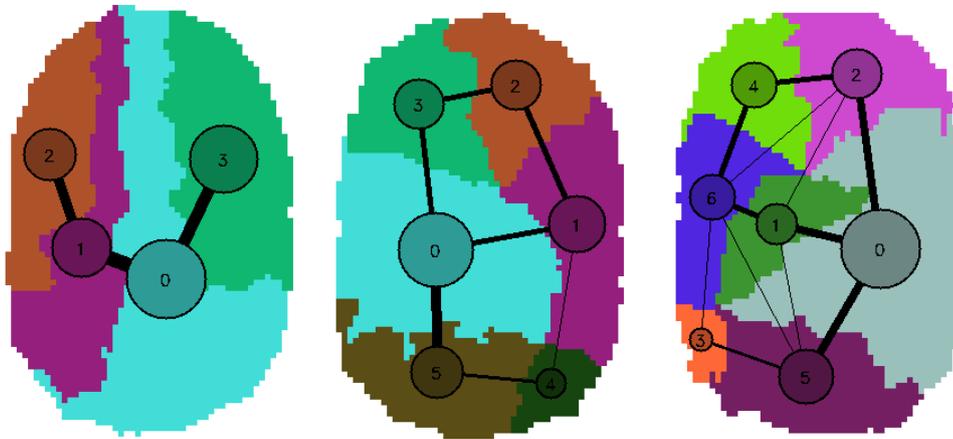


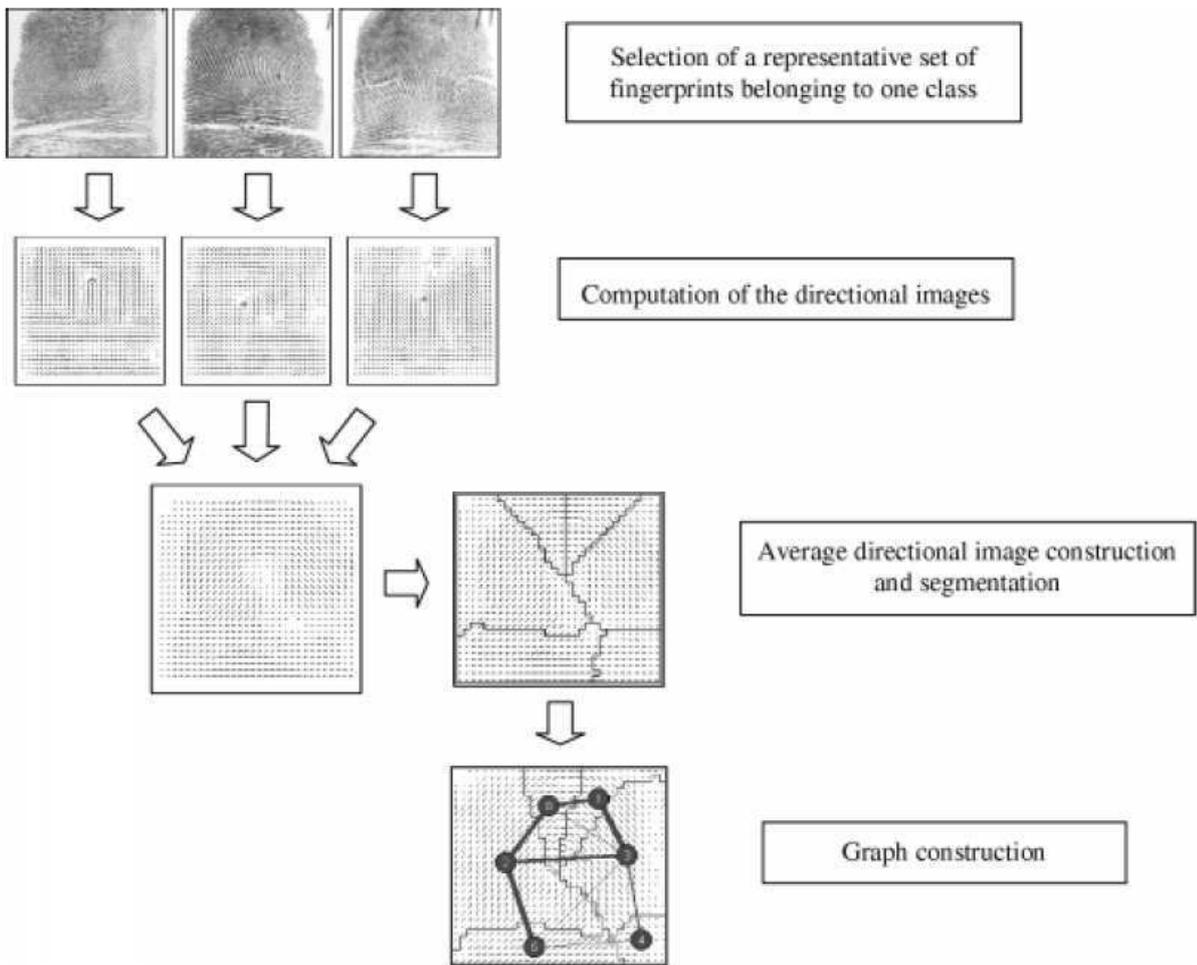
Figura 11 – Exemplos dos grafos associados às segmentações previamente apresentadas. Da esquerda para a direita, tem-se um arco, uma presilha esquerda e uma presilha direita.

## 5.5 Definição de grafos protótipos de referência

Segundo [Maio e Maltoni, 1996](#), a partir de uma análise qualitativa dos grafos obtidos pela segmentação de imagens de orientação, foi possível notar várias similaridades entre grafos de mesma classe, o que sugere que um sistema de classificação pode ser criado para decidir pela classe de uma dada impressão digital a partir do grafo gerado.

Naturalmente, tal sistema enseja a criação de grafos protótipos que sejam representativos de cada classe. Assim, um conjunto adequado de protótipos deve ser criado, no mínimo um grafo por classe.

O procedimento adotado para criação de protótipos é apresentado na Figura 12: são selecionadas impressões digitais representativas de cada classe; constrói-se a imagem de orientação formada pela média das imagens de orientação de cada impressão; o grafo protótipo é gerado a partir da segmentação desta imagem de orientação média.



Fonte: Maltoni et al., 2009

Figura 12 – Procedimento para criação de grafos protótipos.

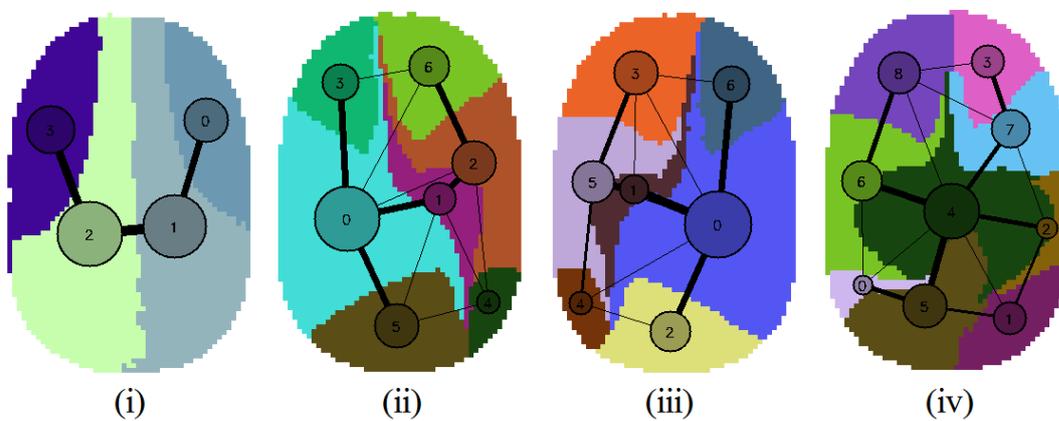


Figura 13 – Protótipos criados para as classes: arco (i), left loop (ii), right loop (iii) e whorl (iv).

A partir do software *Anguli*, um gerador automático de impressões digitais (Nandeo, 2012), foi gerado um banco de dados contendo 100 impressões digitais distintas para cada uma das seguintes classes: arco, presilha esquerda (*left loop*), presilha direita (*right loop*) e verticilo (*whorl*). Com este banco, o procedimento descrito na Figura 12 foi utilizado para gerar os protótipos apresentados na Figura 13.

A geração de protótipos pode ser um tanto empírica, mas alguns características são esperadas para os templates de cada classe: um grafo do tipo arco deve exibir um formato simples de U; já os arcos do tipo presilha (esquerda e direita) são versões espelhadas um do outro, caracterizados por possuírem um arco significativo no lado esquerdo ou direito, respectivamente; por fim, um arco do tipo verticilo tende a possui um nó central e um anel externo. No intuito de criar protótipos que apresentassem tais características, a parametrização do processo de segmentação foi feita de forma individual para cada protótipo.

## 5.6 Casamento inexato de grafos

No reconhecimento de padrões, padrões tendem a ser bastante distorcidos em relação aos seus respectivos protótipos. Assim, não se pode esperar que o casamento de grafos seja tratado como a busca por um isomorfismo “perfeito”. Trata-se, na verdade, de chamado casamento *inexato* de grafos, onde procura-se definir critérios que possam caracterizar a similaridade entre dois grafos.

O método proposto em Lumini, Maio e Maltoni, 1999, apesar de apresentar, no pior caso, complexidade computacional exponencial em relação ao número de nós dos grafos, alcança excelente performance para pequenos grafos (máximo de 10 a 12 nós) a partir da definição de um limitante (*bound*) eficiente. Este método é, portanto, compatível com a natureza dos grafos aqui tratados, isto é, aqueles gerados a partir da segmentação de imagens de orientação, onde o número de nós é pequeno.

### 5.6.1 Definição

O casamento inexato de grafos (*inexact graph matching – igm*) entre dois grafos  $G_1(V_1, A_1, f_{V_1}, f_{A_1})$  e  $G_2(V_2, A_2, f_{V_2}, f_{A_2})$  pode ser definido por uma função *igm* que especifica a atribuição de nós entre os grafos e, conseqüentemente, o mapeamento de seus arcos. Como os grafos, em geral, podem ser formados por um número diferente de nós, é interessante que a função *igm* permita que alguns nós não sejam atribuídos ou, de outra forma, sejam associados a um elemento nulo (nó  $\lambda$ ). A idealização de um nó nulo  $\lambda$  permite tratar a exclusão de um nó como um caso particular de atribuição.

Uma definição formal da função *igm* é dada por:  $igm : V'_1 \rightarrow V'_2$ , com  $V'_1 = V_1 \cup \{\lambda\}$ ,  $V'_2 = V_2 \cup \{\lambda\}$ , *igm* tal que:

- $igm(\lambda) \neq \lambda$
- $\forall v, v' \in V_1$  e  $\forall igm(v), igm(v') \in V_2, v \neq v' \Leftrightarrow igm(v) \neq igm(v')$
- $\forall v_2 \in V_2 \exists v_1 \in V'_1$  tal que  $igm(v_1) = v_2$  e  $\forall v_1 \in V_1 \exists v_2 \in V'_2$  tal que  $igm(v_1) = v_2$

O casamento entre os grafos  $G_1$  e  $G_2$  consiste na transformação deles com o objetivo de tornar-los isomorfos, isto é, são realizadas modificações sobre os grafos para que os mesmos se tornem tão similares quanto possível. Cada modificação feita acarretará em um custo e a similaridade entre os grafos será aferida a partir de quão custoso foi o casamento entre eles.

### 5.6.2 Operações básicas sobre grafos

Os nós  $v \in V_1$  tais que  $igm(v) = \lambda$  são removidos de  $G_1$ , assim como os nós  $v \in V_2$  tais que  $igm(\lambda) = v$  são deletados de  $G_2$ ; nos demais casos, que não envolvem o nó nulo  $\lambda$ , os nós remanescentes  $v \in V_1$  são substituídos por seus nós correspondentes  $igm(v) \in V_2$ . Tais atribuições de nós induzem o mapeamento de arcos. Assim, a deleção de um arco pode ser causada pela ausência de um arco correspondente no outro grafo ou pela deleção anterior de seus nós (deleção forçada); já a substituição de arco é uma consequência natural da associação dos nós que o definem.

As operações básicas envolvidas no casamento de grafos são: substituição de nó *SUBNODE*, deleção de nó *DELNODE*, substituição de arco *SUBARC*, deleção de arco *DELARC* e deleção de arco forçada *DELARC'*. Cada operação possui um custo e, basicamente, operações de deleção são vistas como substituição por elementos (nó ou arco) nulos. O nó nulo  $\lambda$  é definido como um nó que corresponde a uma região isolada (sem regiões adjacentes a ela) com área nula; um arco nulo  $a_{xy}$  é definido como um laço (*self-loop*) com  $x \equiv y$  e  $AdjP(R_x, R_y) = \Delta dir(R_x, R_y) = Bdist(R_x, R_y) = 0$ . A seguir, são definidos os custos associados a cada uma das operações.

- $\forall v_i \in V_1$  e  $v_h \in V_2, SUBNODE(v_i, v_h) = w_N \cdot |\Área(R_i) - \Área(R_h)|$
- $\forall v_x \in V_1$  ou  $v_x \in V_2, DELNODE(v_x) = w_N \cdot \Área(R_x)$
- $\forall a_{ij} \in A_1$  e  $\forall a_{hk} \in A_2,$

$$\begin{aligned}
 SUBARC(a_{ij}, a_{hk}) = & \frac{1}{2} \cdot |AdjP(R_i, R_j) + AdjP(R_h, R_k)| \cdot \\
 & \{w_{A_1} \cdot |AdjP(R_i, R_j) - AdjP(R_h, R_k)| + \\
 & w_{A_2} \cdot \Delta abs(\Delta dir(R_i, R_j), \Delta dir(R_h, R_k)) + \\
 & w_{A_3} \cdot |Bdist(R_i, R_j) - Bdist(R_h, R_k)|\}
 \end{aligned}$$

onde  $\Delta abs(\theta_i, \theta_j) \in [0^\circ, 90^\circ]$  retorna a diferença de fase entre  $\theta_i$  e  $\theta_j \in [-90^\circ, 90^\circ[$ :

$$\Delta abs(\theta_i, \theta_j) = \begin{cases} |\theta_i - \theta_j| & \text{se } |\theta_i - \theta_j| < 90^\circ \\ ||\theta_i - \theta_j| - 180^\circ| & \text{caso contrário} \end{cases} \quad (5.13)$$

- $\forall a_{xy}, a_{xy} \in A_1$  ou  $a_{xy} \in A_2$ ,

$$\begin{aligned} DELARC(a_{xy}) = & \frac{1}{2} \cdot AdjP(R_x, R_y) \cdot \\ & \{w_{A_1} \cdot AdjP(R_x, R_y) + \\ & w_{A_2} \cdot \Delta abs(\Delta dir(R_x, R_y), 0) + \\ & w_{A_3} \cdot Bdist(R_x, R_y)\} \end{aligned}$$

- $\forall a_{xy}, a_{xy} \in A_1$  ou  $a_{xy} \in A_2$ ,  $DELARC'(a_{xy}) = \frac{1}{2} \cdot DELARC(a_{xy})$

Os pesos  $w_N$ ,  $w_{A_1}$ ,  $w_{A_2}$  e  $w_{A_3}$  (positivos) definem a significância de cada atributo e foram escolhidos como segue:  $w_n = 4$ ,  $w_{A_1} = 1.4$ ,  $w_{A_2} = 6.35$  e  $w_{A_3} = 1.4$ .

Basicamente, quanto menor for o custo total das transformações necessárias para que dois grafos se tornem isomorfos, mais similares eles eram inicialmente. Um casamento inexato de grafos  $igm$  tem seu custo total  $C(igm)$  como sendo a soma de todos os custos oriundos das transformações básicas descritas pela função  $igm$ . Naturalmente, para dois grafos, o casamento pode ser feito de várias maneiras diferentes: sendo  $igm_1, igm_2, \dots, igm_n$  todos os possíveis casamentos inexatos entre  $G_1$  e  $G_2$ , o custo do *casamento inexato ótimo de grafos* é dado por:

$$C_{OPT}(G_1, G_2) = \min_{igm_1, igm_2, \dots, igm_n} \{C(igm_i)\} \quad (5.14)$$

Ainda, como os custos associados às operações básicas respeitam as propriedades de uma métrica, isto é, são positivos, simétricos e satisfazem a desigualdade triangular, o custo  $C_{OPT}(G_1, G_2)$  é uma métrica, de modo que  $C_{OPT}(G_1, G_2) = C_{OPT}(G_2, G_1)$ . Como procurar a atribuição ótima testando todas as possibilidades é extremamente exaustivo, um algoritmo de busca em espaço de estados é utilizado.

### 5.6.3 Busca em espaço de estados pelo casamento ótimo

Uma função de casamento  $igm$  pode ser representada por uma matriz binária de atribuição  $X$ , onde o elemento  $x_{ij}$  é igual a 1, caso o nó  $i$  do grafo  $G_1$  tenha sido atribuído ao nó  $j$  de  $G_2$ , ou igual a 0, caso contrário. A matriz  $X$  é similar a uma matriz de permutação, com exceção de que a ela é adicionada uma linha e uma coluna referente ao nó nulo  $\lambda$ .

A determinação do casamento inexato *ótimo igm* entre grafos é baseada em uma busca num espaço de estados. Um estado  $S$  neste espaço de busca corresponde a um casamento parcial entre  $G_1$  e  $G_2$ , onde nem todos os nós foram associados. No estado *inicial* da busca, nenhum nó o foi. Consideremos o exemplo ilustrado na Figura 14, onde os grafos  $G_1$  (à esquerda) e  $G_2$  (à direita) possuem 3 e 4 nós, respectivamente.

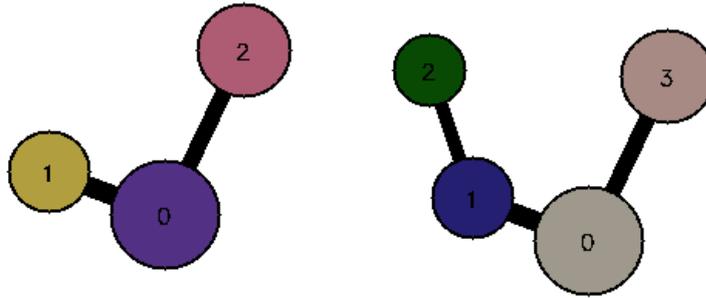


Figura 14 – Exemplo de um cenário de casamento entre dois grafos  $G_1$  e  $G_2$  com 3 e 4 nós, respectivamente.

O estado inicial  $S_0$  pode ser representado pela matriz nula  $X_0$  de dimensões  $4 \times 5$ , o que indica que não houve nenhuma atribuição até então. A matrix  $X$ , de forma geral, deve ter dimensões  $(n_1 + 1) \times (n_2 + 1)$ , onde  $n_1$  e  $n_2$  são os números de nós de  $G_1$  e  $G_2$ , respectivamente. Como mencionado, a linha e coluna adicionais representam o nó nulo  $\lambda$ .

$$X_0 = \left[ \begin{array}{cccc|c} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \end{array} \right] \quad (5.15)$$

Intuitivamente, o leitor pode chegar à conclusão de que a matriz  $X$  correspondente ao casamento ótimo é da seguinte forma:

$$X = \left[ \begin{array}{cccc|c} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 \end{array} \right] \quad (5.16)$$

onde foram feitas as atribuições  $0 \rightarrow 0$ ,  $1 \rightarrow 1$ ,  $\lambda \rightarrow 2$  e  $2 \rightarrow 3$ . De fato, como mostrado na Figura 15, este é o casamento ótimo neste exemplo.

Assim, cabe ao algoritmo caminhar no espaço de buscas, partindo de  $S_0$  e indo até o estado ótimo, se possível. Este “caminhar” é feito um passo (atribuição) por vez. Assim, um novo estado  $S'$  é alcançado a partir da expansão do estado atual  $S$  por meio da atribuição de um nó e, eventualmente, de seus arcos. O estado inicial tem custo nulo e o custo de um estado intermediário (ou final) pode ser obtido a partir do custo do estado

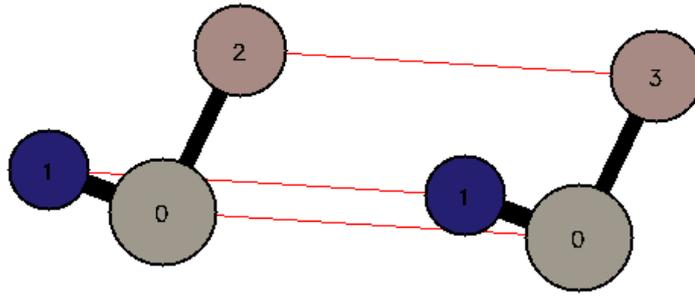


Figura 15 – Resultado do casamento entre os grafos da Figura 14. Custo total do casamento foi de  $C(G_1, G_2) = 60138$ .

anterior somado aos custos de todas as transformações realizados no último passo:

$$C(S') = C(S) + [\text{custo das transformações induzidas pela associação de nó}] \quad (5.17)$$

O estado final corresponde ao casamento inexato *completo*, onde todos os nós foram atribuídos. Em termos da matriz  $X$ , o estado final é alcançado quando as seguintes restrições são satisfeitas:

$$\sum_{i=0}^{n_1} x_{ij} = 1, \quad j = 0, \dots, n_1 - 1 \quad (5.18)$$

$$\sum_{j=0}^{n_2} x_{ij} = 1, \quad i = 0, \dots, n_2 - 1 \quad (5.19)$$

A Figura 16 ilustra como o algoritmo avalia o passo a ser dado no processo de busca. A cada estado  $S$  está associado um limiar inferior (*lower bound* –  $LB(S)$ ), que fornece uma estimativa do custo residual necessário para completar o casamento. Trata-se de uma estratégia do tipo *branch-and-bound* (ramificar-e-limitar) que expande, a cada passo, o estado  $S$  mais promissor, isto é, aquele que apresenta o menor valor de  $C(S) + LB(S)$ .

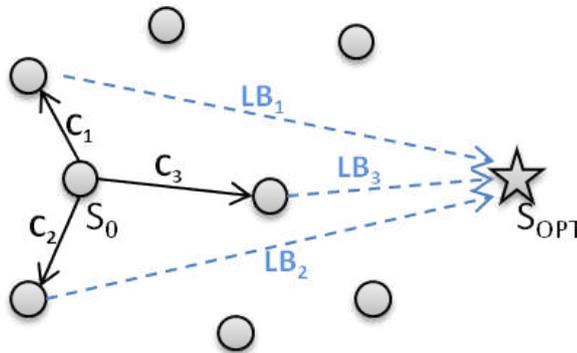


Figura 16 – Diagrama ilustrando a avaliação do primeiro passo na busca utilizando a abordagem *branch-and-bound*.

Fazer uso de um *lower bound* permite que o algoritmo não se engane por observar apenas o custo do estado: na Figura 16, por exemplo, partindo de  $S_0$ , pode-se decidir

por expandir para o estado de custo  $C_1$ , por ser o menor entre  $C_1$ ,  $C_2$  e  $C_3$ , isto é, entre os possíveis estados diferentes de  $S_0$  apenas pela atribuição de um nó. No entanto, a possibilidade de estimar  $LB(S)$  pode indicar que, apesar de pouca custosa, a decisão pode distanciar ainda mais a solução do estado final ótimo. Assim, a escolha mais promissora seria a terceira, com custo total de  $C_3 + LB_3$ .

A busca finaliza quando  $C(S) + LB(S)$  se torna superior que o custo da melhor solução já alcançada. No estado final, haverá sido acumulado todo o custo necessário para o casamento dos grafos, desde o estado inicial. Com isso, tem-se o custo total  $C(G_1, G_2)$ . No caso da Figura 15, por exemplo, o custo total foi de  $C(G_1, G_2) = 60138$ .

Para fins de classificação, é importante que este custo seja normalizado, de modo que seja possível compará-lo com outros casamentos. Vale lembrar que o casamento dos grafos corresponde a uma série de substituições e remoções de nós e arcos com o intuito de torná-los isomorfos. O maior custo possível é alcançado por um casamento onde todos os nós e arcos de ambos os grafos são removidos. Este casamento possui o custo de  $C_{OPT}(G_1, G_\lambda) + C_{OPT}(G_2, G_\lambda)$ , ou seja, a soma dos custos necessários para transformar  $G_1$  e  $G_2$  em isomorfismos do grafo nulo  $G_\lambda$  (um grafo sem qualquer nó ou arco). A normalização é feita, portanto, da seguinte forma:

$$C_{n_{OPT}}(G_1, G_2) = \frac{C_{OPT}(G_1, G_2)}{C_{OPT}(G_1, G_\lambda) + C_{OPT}(G_2, G_\lambda)} \quad (5.20)$$

No exemplo da Figura 15, tem-se que  $C_{OPT}(G_1, G_\lambda) = 362364$  e  $C_{OPT}(G_2, G_\lambda) = 356211$ , o que leva ao resultado  $C_{n_{OPT}}(G_1, G_2) = 60138 / (362364 + 356211) = 0,084$ . Como  $C_{n_{OPT}}(G_1, G_2)$  pertence ao intervalo  $[0, 1]$ , conclui-se que os grafos são bastante similares.

#### 5.6.4 Cálculo do limiar inferior de Gilmore Lawler

O casamento de grafos pode ser formulado como um caso particular de um dos problemas mais difíceis de otimização: o Problema Quadrático de Alocação (*Quadratic Assignment Problem* – *QAP*). A formulação matemática mais utilizada é a de [Koopmans e Beckmann, 1957](#), que modela o caso em que  $n$  instalações devem ser alocadas em  $n$  localizações; o custo de uma alocação é definido de acordo com a quantidade de fluxo existente entre cada par de instalações multiplicada pela distância entre as localizações correspondentes.

Em se tratando de casamento de grafos, as instalações podem ser vistas como os nós do primeiro grafo, enquanto que as localizações seriam os nós do segundo grafo. O custo deixa de ter a interpretação de fluxo-distância para ser baseado nos parâmetros dos arcos, isto é, nos perímetros de adjacência, diferença de fase e distância euclidiana. Ainda, nesta nova interpretação, é preciso incorporar a ideia de que os grafos podem ter números diferentes de nós. Portanto, nós nulos podem ser adicionados para levar em conta deleções.

Sejam  $n_1$  e  $n_2$  os números de nós dos grafos  $G_1$  e  $G_2$ , respectivamente. Cada nó no primeiro grafo deve ser associado a, no máximo, um nó do segundo e vice-versa. A associação de nós implicitamente induz a de arcos.

Seja  $E = [e_{ih}]$ ,  $i, h = 0, \dots, n_1 - 1$  uma matriz referente a  $G_1$  que codifica os atributos dos arcos  $a_{ih} \in A_1$ . No contexto original, a matriz  $E$  seria formada pelas distâncias entre cada par de localizações, mas neste contexto, deve também levar em conta os demais atributos dos arcos. Assim, cada elemento  $e_{ih}$  é definido como:

$$e_{ih} = w_{A_1} \cdot AdjP(R_i, R_h) + w_{A_2} \cdot \Delta abs(\Delta dir(R_i, R_h), 0) + w_{A_3} \cdot Bdist(R_i, R_h) \quad (5.21)$$

De forma análoga, a matriz  $F = [f_{jk}]$ ,  $j, k = 0, \dots, n_2 - 1$  refere-se a  $G_2$  e codifica os atributos do arcos  $a_{jk} \in A_2$ . No contexto original,  $F$  seria formada pelos fluxos entre cada par de instalações, mas aqui é definida de forma similar à matriz  $E$ . O elemento  $f_{jk}$  é dado por:

$$f_{jk} = w_{A_1} \cdot AdjP(R_j, R_k) + w_{A_2} \cdot \Delta abs(\Delta dir(R_j, R_k), 0) + w_{A_3} \cdot Bdist(R_j, R_k) \quad (5.22)$$

Nota-se, portanto, que as matrizes  $E$  e  $F$  dizem respeito aos *arcos* dos grafos. Para também levar em conta os *nós*, define-se a matriz  $C = [c_{ij}]$ ,  $i = 0, \dots, n_1 - 1$ ,  $j = 0, \dots, n_2 - 1$ , que codifica o custo para associar o nó  $i$  de  $G_1$  ao nó  $j$  de  $G_2$ .

Ainda, a matriz  $X = [x_{ij}]$ ,  $i = 0, \dots, n_1 - 1$ ,  $j = 0, \dots, n_2 - 1$ , como já foi apresentado, assume valor 1 se o nó  $i$  está associado ao nó  $j$  ou 0 caso contrário. Com isso, formula-se o *QAP*:

$$\mathbf{z}_{QAP} = \min z = \sum_{i=0}^{n_1-1} \sum_{j=0}^{n_2-1} \sum_{h=0}^{n_1-1} \sum_{k=0}^{n_2-1} e_{ih} f_{jk} x_{hk} x_{ij} + \sum_{i=0}^{n_1-1} \sum_{j=0}^{n_2-1} c_{ij} x_{ij} \quad (5.23)$$

sujeito às seguintes restrições:

$$\sum_{i=0}^{n_1} x_{ij} = 1, \quad j = 0, \dots, n_2 - 1 \quad (5.24)$$

$$\sum_{j=0}^{n_2} x_{ij} = 1, \quad i = 0, \dots, n_1 - 1 \quad (5.25)$$

$$x_{ij} \in \{0, 1\}, \quad i = 0, \dots, n_1 - 1, \quad j = 0, \dots, n_2 - 1 \quad (5.26)$$

A Equação 5.23 representa um problema de otimização cujo objetivo é encontrar a matriz  $X$  que a minimiza. A primeira parcela da expressão (composta de quatro somatórios) está associada à atribuição de arcos: de quantas maneiras é possível atribuir cada arco de  $G_1$  a um arco de  $G_2$ ? A resposta para esta pergunta envolve quatro graus de liberdade: de quantas maneiras pode-se escolher um nó  $i$  em  $G_1$  e de quantas maneiras este nó pode ser mapeado em  $G_2$  (escolha de  $j$ ); ao mesmo tempo, de quantas maneiras pode-se escolher

um segundo nó  $h$  em  $G_1$  (formando o arco  $ih$ ) e de quantas maneiras o nó  $h$  pode ser mapeado em  $G_2$  (escolha de  $k$ , formando o arco  $hk$ )?

É muito difícil resolver este problema de otimização diretamente pela minimização da Equação 5.23, de forma que maneiras alternativas são propostas para, por exemplo, decompor o problema em subproblemas de maior tratabilidade. Limiares inferiores (*lower bounds*) constituem parte destas alternativas e o melhor limiar conhecido para o QAP é o apresentado por Gilmore e Lawler. Para induzir o limiar, consideremos reescrever a Equação 5.23 da seguinte forma:

$$z_{QAP} = \min z = \sum_{i=0}^{n_1-1} \sum_{j=0}^{n_2-1} \left[ \sum_{h=0}^{n_1-1} \sum_{k=0}^{n_2-1} e_{ih} f_{jk} x_{hk} \right] \cdot x_{ij} + \sum_{i=0}^{n_1-1} \sum_{j=0}^{n_2-1} c_{ij} x_{ij} \quad (5.27)$$

onde a expressão entre colchetes apresenta  $i$  e  $j$  fixos. Assim, o QAP pode ser decomposto em vários problemas menores, um para cada par  $(i, j)$ , a partir da minimização da expressão entre colchetes. Supondo que já foram escolhidos o nó  $i$  de  $G_1$  e seu correspondente  $j$  em  $G_2$ , de quantas maneiras é possível escolher o nó  $h$  em  $G_1$  e o nó  $k$  em  $G_2$ , mapeando o arco  $a_{ih}$  no arco  $a_{jk}$ ? O problema é, portanto, subdividido em vários problemas de alocação linear (*Linear Assignment Problems - LAP*).

Para calcular o limiar inferior de Gilmore Lawler, deve-se primeiramente calcular, para cada par  $(i, j)$ ,  $i = 0, \dots, n_1 - 1$ ,  $j = 0, \dots, n_2 - 1$ , o valor de  $z_{ij}$  dado por:

$$z_{ij} = \min \sum_{h=0}^{n_1-1} \sum_{k=0}^{n_2-1} e_{ih} f_{jk} x_{hk} \quad (5.28)$$

sujeito às mesmas restrições nas variáveis  $x_{hk}$ . Uma vez calculados, os termos permitem a construção do limiar inferior de *Gilmore-Lawler*, dado por:

$$z_{GLB} = \min z = \sum_{i=0}^{n_1-1} \sum_{j=0}^{n_2-1} [z_{ij}] \cdot x_{ij} + \sum_{i=0}^{n_1-1} \sum_{j=0}^{n_2-1} c_{ij} x_{ij} \quad (5.29)$$

$$z_{GLB} = \min z = \sum_{i=0}^{n_1-1} \sum_{j=0}^{n_2-1} (z_{ij} + c_{ij}) x_{ij} \quad (5.30)$$

A princípio, os coeficientes  $z_{ij}$  levariam, cada um, à resolução de um problema linear (*LAP*). No entanto, sua minimização pode ser facilmente realizada a partir da ordenação dos elementos dos vetores linha  $e_{ih}$  ( $i$  fixo) e  $f_{jk}$  ( $j$  fixo) ambos em ordem ascendente ou descendente e, então, do cálculo de seu produto interno. Isto associará os arcos fortes de  $G_1$  aos de  $G_2$ , bem como os mais fracos de  $G_1$  aos de  $G_2$ . Construir a matriz  $Z$  desta maneira reduz a complexidade computacional do *GLB* para  $O(n^3)$ .

A atribuição de um nó  $i$  a um nó  $j$  acarretará em um mapeamento de arcos com custo de, no mínimo,  $z_{ij}$ . Por fim, a Equação 5.30 configura um problema de alocação linear com uma matriz de custo formado pelos elementos  $(z_{ij} + c_{ij})$ , isto é, levando em conta o peso de nós e de arcos. Esta equação mostra que a atribuição de um nó  $i$  a um

nó  $j$  custa  $c_{ij}$ , além de um adicional mínimo de  $z_{ij}$  referente aos eventuais arcos. Para minimizar a Equação 5.30, fez-se uso do Método Húngaro, um algoritmo clássico para resolução de um LAP (Kuhn, 1955).

- **Expandindo as matrizes  $C$  e  $Z$  para incorporação do nó nulo**

Em termos da matriz  $X$ , como discutido anteriormente, a incorporação do nó nulo  $\lambda$  se dá pela inclusão de uma nova linha e coluna para o mesmo. Já as matrizes  $C$  e  $Z$  são expandidas não por uma única linha e coluna, mas por blocos diagonais. Na matriz  $C$ , dada por

$$C = \left[ \begin{array}{ccc|ccc} SUBN(R_0^1, R_0^2) & \dots & SUBE(R_0^1, R_{n_2-1}^2) & DELN(R_0^1) & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ SUBN(R_{n_1-1}^1, R_0^2) & \dots & SUBN(R_{n_1-1}^1, R_{n_2-1}^2) & 0 & \dots & DELN(R_{n_1-1}^1) \\ \hline DELN(R_0^2) & \dots & 0 & \infty & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & DELN(R_{n_2-1}^2) & 0 & \dots & \infty \end{array} \right]$$

nota-se que o bloco superior esquerdo corresponde aos custos de substituição de nós, conforme previstos. Já nos blocos superior direito e inferior esquerdo são dispostos os custos para remoção de nós do grafo  $G_1$  e  $G_2$  ao longo de suas diagonais. Na prática, esta medida equivale à existência de múltiplos nós nulos, o que permite que mais de um nó seja removido sem violar as restrições de múltiplas associações em uma linha ou coluna. Por último, o bloco inferior direito tem custos infinitos, o que evita que nós nulos sejam mapeados em nós nulos.

A formação da matriz  $Z$  é similar: no seu bloco superior esquerdo são dispostos os elementos  $z_{ij}$  definidos pela Equação 5.28. Nos blocos superior direito e inferior esquerdo, são dispostos os custos referentes a deleção forçada de todos os arcos associados ao nó removido.

$$Z = \left[ \begin{array}{ccc|ccc} z_{0,0} & \dots & z_{0,n_2-1} & DELA_0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ z_{n_1-1,0} & \dots & z_{n_1-1,n_2-1} & 0 & \dots & DELA_{n_1-1} \\ \hline DELA_0 & \dots & 0 & \infty & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & DELA_{n_2-1} & 0 & \dots & \infty \end{array} \right]$$

## 5.6.5 Um exemplo

A seguir são apresentados os resultados dos casamentos de um grafo de teste com cada os protótipos. O grafo de teste corresponde à impressão digital do tipo arco da Figura 10(i).

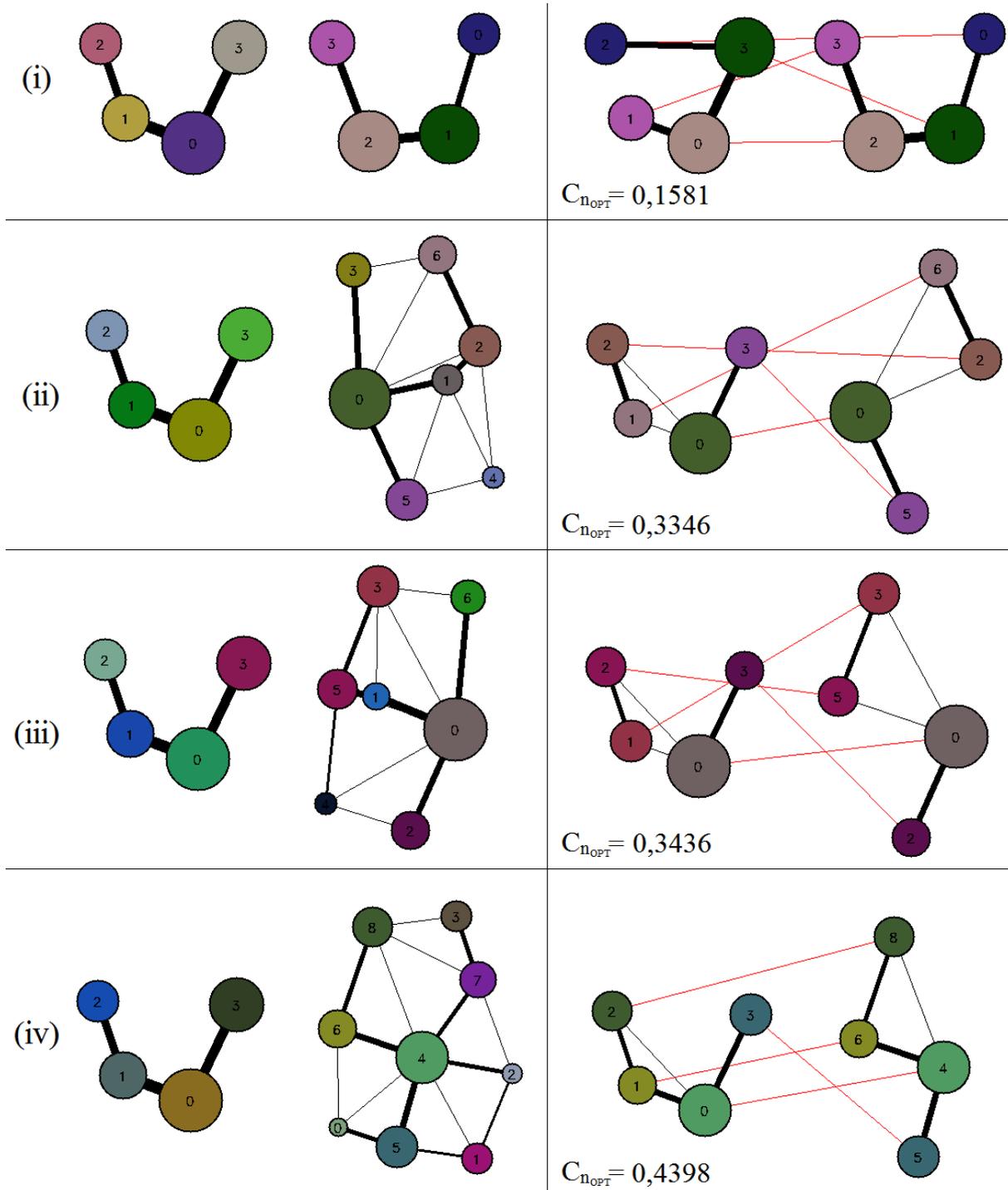


Figura 17 – Casamento de um grafo oriundo de uma classe tipo arco com os protótipos das classes: (i) arco, (ii) presilha esquerda (left loop), (iii) presilha direita (right loop) e (iv) verticilo (whorl). No lado esquerdo, os grafos são apresentados em seus estados iniciais e no lado direito após os respectivos casamentos.

## 5.7 Classificação

O casamento de um grafo qualquer  $G$ , no contexto das quatro classes prototipadas (*arco* –  $A$ , *left loop* –  $LL$ , *right loop* –  $RL$  e *whorl* –  $W$ ), deve retornar um vetor de distâncias do tipo:

$$\mathbf{w}_G = [C_{nOPT}(G, P_A), C_{nOPT}(G, P_{LL}), C_{nOPT}(G, P_{RL}), C_{nOPT}(G, P_W)] \quad (5.31)$$

Para o exemplo da Figura 17 o vetor obtido foi:

$$\mathbf{w}_G = [0,1581 \quad 0,3346 \quad 0,3436 \quad 0,4398] \quad (5.32)$$

O critério mais simples que pode ser adotado para classificar o grafo é pela distância mínima. A princípio, este é o critério adotado para a classificação. Posteriormente, um classificador mais robusto poderá ser incorporado ao sistema. Por esse critério, o grafo do exemplo seria corretamente classificado, visto que está mais próximo do protótipo de arco que dos demais.

Com a implementação do algoritmo por completo, foi realizado o primeiro teste preliminar do classificador. Com o mesmo banco de dados utilizado para criação dos protótipos (um total de 400 impressões digitais, sendo 100 de cada classe) o primeiro teste do classificador foi realizado. O resultado obtido foi a matriz de confusão apresentada na Figura 18.

		GRAFO PROTÓTIPO			
		ARCH	LEFT LOOP	RIGHT LOOP	WHORL
GRAFO TESTE	%				
	ARCH	75	10	15	0
	LEFT LOOP	0	21	12	67
	RIGHT LOOP	4	18	55	23
WHORL	0	1	0	99	

Figura 18 – Matriz de confusão resultante do primeiro teste.

Como pode-se perceber, os resultados obtidos foram muitos bons para as classes verticilo (*whorl*) e arco (*arch*), mas se mostraram bastante rudimentares nas classes de presilha esquerda e direita (*left/right loop*). Novamente, este foi o primeiro teste realizado e há muitos aspectos que podem ser abordados e aprimorados: melhorar a parametrização dos algoritmos de segmentação e de casamento, aumentar o número de impressões digitais, investigar a criação dos protótipos, podendo-se utilizar mais de um protótipo por classe, entre outras possibilidades.



## 6 Conclusão

Durante este estágio, foi possível entrar em contato com a área de processamento biométrico, no contexto de uma empresa que vem se especializando na área, e aplicar o conhecimento prévio de processamento digital de imagens. A divisão de pesquisa da VSoft oferece um ambiente amigável para discussão e troca de ideias.

O estágio realizado se voltou para a implementação de um sistema de classificação de impressões digitais, que visa contribuir com o reconhecimento das impressões a partir da definição de sua classe. No tempo hábil do estágio, o sistema foi implementando, encontrando-se agora em estado de revisão e aprimoramento.

Como apresentado neste relatório, a performance do classificador ainda pode melhorar bastante e, para isso, algumas abordagens podem ser tomadas, tais como: o aumento no número de protótipos, bem como o seu aprimoramento, uma nova parametrização dos algoritmos envolvidos no sistema de classificação, mudanças no banco de impressões digitais utilizados ou, ainda, uma reformulação de certos aspectos como, por exemplo, a proposição de novas funções de custos.



# Referências

- ITSEEZ. *The OpenCV Reference Manual*. 2.4.9.0. ed. OpenCV. Citado na página [37](#).
- KOOPMANS, T. C.; BECKMANN, M. Assignment problems and the location of economic activities. *Econometrica: journal of the Econometric Society*, JSTOR, p. 53–76, 1957. Citado na página [50](#).
- KUHN, H. W. The hungarian method for the assignment problem. *Naval research logistics quarterly*, Wiley Online Library, v. 2, n. 1-2, p. 83–97, 1955. Citado na página [53](#).
- LUMINI, A.; MAIO, D.; MALTONI, D. Inexact graph matching for fingerprint classification. *Machine Graphics and Vision*, v. 8, p. 231–248, 1999. Citado 4 vezes nas páginas [23](#), [35](#), [39](#) e [45](#).
- MAIO, D.; MALTONI, D. A structural approach to fingerprint classification. In: IEEE. *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*. IEEE, 1996. v. 3, p. 578–585. Citado 2 vezes nas páginas [40](#) e [43](#).
- MALTONI, D. et al. *Handbook of fingerprint recognition*. Publicado por: Springer Science & Business Media, 2009. Citado 7 vezes nas páginas [21](#), [29](#), [30](#), [31](#), [32](#), [36](#) e [44](#).
- NAMDEO, J. S. Generating, classifying and indexing large scale fingerprints. *Indian Institute of Science*, 2012. Citado 2 vezes nas páginas [37](#) e [45](#).