

Rômulo Ferreira Douro

***Reconhecimento de Formas em Imagens Através da
Associação de Pontos e Avaliação Multicritério de
Arestas***

Vitória, ES – Brasil

27 de agosto de 2012

Rômulo Ferreira Douro

***Reconhecimento de Formas em Imagens Através da
Associação de Pontos e Avaliação Multicritério de
Arestas***

Dissertação apresentada como requisito parcial para obtenção do Grau de Mestre em Ciência da Computação pela Universidade Federal do Espírito Santo.

Orientadora Maria Claudia Silva Boeres

DEPARTAMENTO DE INFORMÁTICA
CENTRO TECNOLÓGICO
UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

Vitória, ES – Brasil

27 de agosto de 2012

Dissertação apresentada como requisito parcial para obtenção do Grau de Mestre em Ciência da Computação sob o título *“Reconhecimento de Formas em Imagens Através da Associação de Pontos e Avaliação Multicritério de Arestas”*, defendida por Rômulo Ferreira Douro em 27 de agosto de 2012, em Vitória, Estado do Espírito Santo, pela banca examinadora constituída pelos professores:

Profa. Dra. Maria Claudia Silva Boeres
Orientadora

Prof. Dr. Roberto Marcondes Cesar Junior
IME-USP

Profa. Dra. Maria Cristina Rangel
Universidade Federal do Espírito Santo

Dedico esse trabalho à minha família.
Ao meu falecido avô, Anacleto, obrigado pelas conversas na varanda.

Agradecimentos

Agradeço a Deus, por me amparar quando estive sozinho.

Agradeço às pessoas que estiveram comigo nessa jornada meus pais, minha irmã, meus amigos e professores e, em especial, minha orientadora, por me apoiar mesmo quando eu não acreditei que fosse conseguir.

*“O conhecimento é a pequena porção da ignorância
que arrumamos e classificamos.”*

— AMBROSE BIERCE

Resumo

A tarefa de reconhecer formas em imagens é uma atividade inserida na área da visão computacional e está diretamente relacionada ao problema de reconhecimento de padrões. Reconhecer um padrão em uma imagem pode ser viabilizado por meio da utilização de grafos como estruturas matemáticas para representação de suas características. O processo de reconhecer uma imagem em outra pode ser executado por meio do casamento entre os grafos que as representam. Neste trabalho as formas presentes em imagens serão representadas por grafos e o processo de associação entre elas será modelado como um problema de atribuição quadrática. Além disso, são propostas técnicas de avaliação para custear a contribuição das associações entre vértices e arestas. São apresentadas também duas adaptações empregadas sobre a técnica Shape Context, além de um método de comparação de grafos baseado em busca em profundidade usando uma avaliação multicritério das características de arestas. A aplicação dessas técnicas a grafos de linha gerados a partir dos grafos originais a serem comparados, é proposta, apresentando resultados promissores. As técnicas aqui implementadas são submetidas a testes aplicados sobre quatro bases de imagens utilizadas em diversos trabalhos existentes na literatura.

Abstract

The task of recognizing shapes in images is an activity located in the area of computer vision and is directly related to the problem of pattern recognition. Recognize a pattern in an image can be made possible through the use of graphs for the representation of their features, since the process of recognizing two images can be run through the matching of the graphs that represent them. In this work, images shapes will be represented by graphs and their matching process is modeled as a quadratic assignment problem. In addition, evaluation techniques are proposed to cost the contribution of the vertices and edges associations. We propose two adaptations on the Shape Context technique and also a matching method based on depth search using a multicriteria evaluation on the characteristics of edges. The application of these techniques to line graphs generated from the original matching graphs is proposed, with promising results. The techniques implemented here are submitted to tests applied on four image databases used in several papers in the literature.

Sumário

Lista de Abreviaturas e Termos

Lista de Símbolos

Lista de Figuras

Lista de Tabelas

1	Introdução	p. 19
1.1	Associação de formas e o reconhecimento de padrões de pontos	p. 20
1.2	Objetivos	p. 21
1.3	Organização do trabalho	p. 23
2	Revisão Bibliográfica	p. 24
2.1	O reconhecimento de padrões	p. 24
2.1.1	O processo de reconhecimento de um padrão	p. 26
2.1.1.1	Pré-processamento	p. 27
2.1.1.2	Extração de características	p. 28
2.1.1.3	Classificação	p. 29
2.2	Problemas abordados em imagens	p. 31
2.2.1	Segmentação de imagens	p. 31
2.2.2	Correspondência de imagens	p. 33
2.3	Correspondência de Imagens via correspondência de grafos	p. 35
2.3.1	Representação de uma imagem como um grafo	p. 36

2.3.1.1	Definições básicas sobre grafos	p. 36
2.3.1.2	Grafos de atributos relacionais (ARG)	p. 37
2.3.2	Formas de abordagem para resolução de problemas de associação em imagens	p. 37
2.3.2.1	Comparação de uma imagem com parte de outra imagem . .	p. 38
2.3.2.2	Comparação de duas imagens	p. 38
2.3.2.3	Comparação de uma imagem com um banco de imagens . . .	p. 39
3	O Problema de Associação de Grafos	p. 41
3.1	Correspondência exata	p. 41
3.2	Correspondência inexata	p. 42
3.3	O reconhecimento de formas	p. 44
3.4	O problema de associação de pontos - PPM	p. 46
3.5	Modelagem do problema	p. 47
3.5.1	Modelagem para o problema de associação de pontos através da as- sociação de grafos	p. 48
4	Estratégias de Resolução	p. 52
4.1	Representação da associação entre grafos	p. 52
4.2	Custo da associação entre grafos	p. 53
4.3	Funções de avaliação	p. 54
4.3.1	Contribuição de vértices - Termo linear	p. 55
4.3.1.1	<i>Shape Context</i> Percentual - <i>SCP</i>	p. 55
4.3.1.2	<i>Shape Context</i> para arestas - <i>SCA</i>	p. 57
4.3.2	Comparação de arestas usando busca em profundidade	p. 64
4.3.3	Avaliação multicritério para arestas	p. 67
4.3.3.1	Critério local para avaliação de arestas - c_{eLocal}	p. 68
4.3.3.2	Critério global para avaliação de arestas - $c_{eGlobal}$	p. 70

Critério de avaliação global 01 - ($c_{eGlobal_01}$):	p. 71
Critério de avaliação global 02 - ($c_{eGlobal_02}$):	p. 72
Cálculo do critério de avaliação global	p. 75
4.3.3.3 Cálculo para o custo da avaliação multicritério de arestas - c'_e	p. 75
4.3.4 Contribuição das arestas - Termo quadrático	p. 76
4.3.4.1 Cálculo da contribuição de arestas - c_e	p. 76
4.4 Método de associação	p. 77
4.4.1 Geração da solução inicial	p. 77
4.4.2 Procedimento de otimização	p. 81
4.4.2.1 Adaptação da heurística <i>2-opt</i> para melhoramento da solução do PPM	p. 82
4.4.3 Otimização para o tempo de execução	p. 85
4.4.4 Geração da solução final - melhor associação entre grafos	p. 87
4.5 O PPM definido sobre os grafos de linha de G_i e G_m	p. 89
4.5.1 Aquisição de informações no grafo de linha	p. 91
4.5.2 Cálculo da contribuição de vértices nos grafos de linha	p. 92
5 Testes Computacionais	p. 94
5.1 Instâncias de testes - Bancos de imagens	p. 94
5.1.1 KIMIA e MPEG96	p. 94
5.1.2 COIL	p. 96
5.1.3 MNIST	p. 97
5.2 Testes efetuados	p. 98
5.2.1 Testes 01 - KIMIA e MPEG96	p. 98
5.2.1.1 Parâmetros de execução	p. 98
5.2.1.2 Metodologia de teste	p. 99
5.2.1.3 Resultados computacionais	p. 99

5.2.2	Testes 02 - KIMIA e MPEG96	p. 101
5.2.2.1	Parâmetros de execução	p. 102
5.2.2.2	Metodologia de teste	p. 102
5.2.2.3	Resultados computacionais	p. 102
5.2.3	Testes 03 - COIL	p. 104
5.2.3.1	Parâmetros de execução	p. 104
5.2.3.2	Metodologia de teste	p. 104
5.2.3.3	Resultados computacionais	p. 105
5.2.4	Testes 04 - MNIST	p. 110
5.2.4.1	Parâmetros de execução	p. 111
5.2.4.2	Metodologia de testes	p. 112
5.2.4.3	Resultados computacionais	p. 112
5.2.4.4	Testes para a distância mínima de associação	p. 112
6	Conclusões e Trabalhos Futuros	p. 116
	Referências	p. 119

Lista de Abreviaturas e Termos

ARG	<i>Attributed Relational Graph</i> - Grafo de Atributos Relacionais
HG	<i>Hashing Geométrico</i>
MCS	<i>Maximum Common Subgraph</i> - Máximo Subgrafo Comum
PPM	<i>Point Pattern Matching</i> - Problema de Casamento de Padrões de Pontos
SCA	<i>Shape Context</i> de Aresta
SCP	<i>Shape Context</i> Percentual

Lista de Símbolos

G_i	Grafo de entrada
G_m	Grafo do modelo
V_i	Conjunto de vértices do grafo de entrada
V_m	Conjunto de vértices do grafo do modelo
E_i	Conjunto de arestas do grafo de entrada
E_m	Conjunto de arestas do grafo do modelo
v_i	Vértice do grafo de entrada
v_m	Vértice do grafo do modelo
e_i	Aresta do grafo de entrada
e_m	Aresta do grafo do modelo

Lista de Figuras

1	Dois conjuntos de pontos e as relações entre eles	p. 21
2	Processo de associação	p. 22
3	Fluxo de trabalho para reconhecimento de padrões	p. 25
4	Sistema de reconhecimento de padrões. Fonte: [1]	p. 26
5	Exemplo de pré-processamento	p. 28
6	Extração de características - Exemplo	p. 29
7	Regra KNN. Fonte: [2]	p. 30
8	Procedimento básico de segmentação de uma imagem	p. 33
9	Associação de imagens que representam grupos de proteínas. Fonte: [3]	p. 34
10	Associação de imagens - Exemplo	p. 34
11	Exemplos de dígitos manuscritos, ambos representando o número dois	p. 35
12	Exemplo de comparação. A imagem 12(a) está na imagem 12(b)?	p. 38
13	Resposta esperada para a presença ou não de uma imagem em outra	p. 38
14	Exemplo de comparação. A imagem 14(a) é mais semelhante a 14(b) ou a 14(c) .	p. 39
15	Exemplo de comparação. A imagem 15(a) é comparada com as imagens do banco 15(b)	p. 40
16	Exemplo de isomorfismo. G_a é isomorfo a G_b	p. 42
17	Exemplo de isomorfismo de subgrafo. G_c , subgrafo de G_a , é isomorfo a G_b . .	p. 43
18	Obtenção do grafo interior a uma forma	p. 44
19	Duas formas e os grafos formados pelos seus eixos médios. Fonte: [4]	p. 45
20	Exemplo de reconhecimento de padrões de pontos	p. 46

21	Analogia: 21(a) é encarado como G_i e deve ser associado a 21(b) que é encarado como G_m	p. 49
22	Analogia: custos de implantação da instalação nas localidades referentes . . .	p. 50
23	Exemplo de associações geradas	p. 53
24	Exemplo de associações com penalização	p. 54
25	Gráfico log-polar	p. 55
26	Exemplo para cálculo de C_{ij} : a imagem em 26(a) representa tomada de pontos para o histograma de um ponto de G_i e a imagem em 26(b) se refere à tomada de pontos em G_m	p. 56
27	A imagem em 26(a) gera o histograma 27(a) e a imagem em 26(b) gera o histograma 27(b)	p. 57
28	Tomada de base - HG. Fonte:[5], com adaptação	p. 58
29	Arestas de entrada e do modelo com suas respectivas vizinhas	p. 59
30	Comparação de arestas - HG	p. 59
31	Perpendicularidade dos eixos do grafo log-polar	p. 60
32	Exemplo de grafo	p. 60
33	Diferenças da técnica original e a adaptada	p. 61
34	Tomada de dados SCA, aresta 01	p. 62
35	Tomada de dados SCA, aresta 02	p. 62
36	Exemplo de tomada de SC	p. 63
37	Exemplo de tomada de SC - adaptado	p. 64
38	Vértices selecionados para comparação v_{i0} e v_{m0}	p. 67
39	Arestas a serem comparadas - Vértices v_{i0} e v_{m0}	p. 68
40	Comparação de arestas em níveis	p. 69
41	Arestas a serem comparadas - c_{eLocal}	p. 69
42	Ângulos entre e_i e e_m - c_{eLocal}	p. 70
43	Comprimentos das arestas - c_{eLocal}	p. 70

44	Aquisição de informações globais em níveis	p. 71
45	Informações globais para comparação no nível 1	p. 72
46	Valores possíveis para o fator angular global 02	p. 73
47	Tomada de fator angular global 02	p. 74
48	Tomada de fator linear global 02	p. 74
49	Exemplo de movimento <i>2-opt</i>	p. 82
50	Exemplo da adaptação <i>2-opt</i> para nosso trabalho	p. 84
51	Grafos de entrada e do modelo	p. 86
52	Avaliação da distância mínima de associação	p. 87
53	Fluxograma representando o esquema de geração das associações entre vértices de G_i e G_m	p. 88
54	Um grafo G e seu correspondente $L(G)$	p. 91
55	SCA para o vértice 1 em $L(G)$	p. 91
56	Segundo valor de SCA para o vértice 1 em $L(G)$	p. 92
57	Banco de imagens KIMIA	p. 95
58	Dificuldades encontradas em KIMIA - Rotação	p. 95
59	Dificuldades encontradas em KIMIA - Rotação	p. 95
60	Dificuldades encontradas em KIMIA - Escalonamento	p. 96
61	Imagens de MPEG. Fonte: [6]	p. 96
62	Tomada de imagens da base COIL	p. 97
63	Imagens do banco MNIST	p. 97
64	Associações obtidas em KIMIA	p. 100
65	Comportamento do algoritmo	p. 101
66	Comportamento do algoritmo	p. 103
67	Taxa de erros em COIL com variação no número de protótipos	p. 105
68	Erro obtido em COIL	p. 106

69	Erro obtido em COIL	p. 107
70	Erro obtido em COIL	p. 107
71	Erro obtido em COIL	p. 108
72	Erro obtido em COIL	p. 108
73	Protótipos usados em COIL	p. 109
74	Protótipos usados em COIL	p. 110
75	Protótipos usados em COIL	p. 111
76	Taxa de erros em MNIST com variação no número de protótipos	p. 113
77	Variação nos tempos de execução com diferentes d_{ma} 's	p. 115

Lista de Tabelas

1	Exemplo de matriz de associação	p. 53
2	Resultados para KIMIA - 10 melhores classificações para cada imagem	p. 99
3	Resultados para MPEG 96 - 11 melhores classificações para cada imagem . .	p. 100
4	Resultados para KIMIA - 10 melhores classificações para cada imagem	p. 103
5	Resultados para MPEG 96 - 11 melhores classificações para cada imagem . .	p. 103
6	Erros encontrados na execução para MNIST. Temos a imagem e acima dela seu rótulo (a esquerda) e a nossa classificação (a direita).	p. 114
7	Tempos de execução com variação da distância mínima de associação	p. 115

1 Introdução

Reconhecer um padrão em uma imagem é uma tarefa relevante na atualidade em que sistemas inteligentes tentam constatar, de forma automatizada, a existência ou não de determinada característica em uma dada imagem. Nos referimos à imagem por ser a base de concentração para a pesquisa a ser realizada neste trabalho, não obstante, o reconhecimento de padrões não se restringe a apenas o arcabouço dessa atividade em especial.

O Reconhecimento de Padrões se apresenta como uma importante área cuja menção se encontra destacada em diversos segmentos de pesquisa como medicina, engenharia e segurança pública. Dentre alguns trabalhos que denotam a amplitude dos temas relacionados a esta área podemos citar aqueles que tratam do reconhecimento de faces, como nos trabalhos de [1], [7] e [8], sendo este último aplicando à perícia criminal, e os que visam a detecção de objetos em geral, como em [9].

Em [10], o autor destaca que o Reconhecimento de Padrões é um problema que se encontra em aberto, apesar de diversas implementações eficientes construídas atualmente. Uma área correlacionada é a Visão Computacional, a qual visa adquirir informações de forma automatizada através da extração de dados de imagens ou vídeo auxiliando em tarefas de tomada de decisão, conforme [11] apud [10]. O trabalho de [12] é um exemplo do uso do reconhecimento de padrões na tomada de decisões onde é usado como forma de detecção de áreas para irrigação visando acompanhar a expansão da atividade da agricultura irrigada no Brasil.

Na visão computacional, o reconhecimento de objetos é um campo de relevante interesse e dentro desse contexto se apresenta o problema do reconhecimento de padrões de pontos, cujo objetivo é apresentar o mapeamento de um conjunto de pontos no outro, o que viabiliza o reconhecimento de formas dos objetos, como em [3]. Tal mapeamento é estabelecido avaliando as características mais importantes encontradas em cada par de objetos, dado que os pontos são associados a essas características, conforme denota [3] onde o autor ainda expõe a grande variedade de aplicações do problema relacionado como a associação

de imagens de eletroforese bidimensional e para reconhecimento de dígitos manuscritos.

No âmbito das atividades que visam o reconhecimento de objetos encontra-se o Casamento de Formas, ou *Shape Matching*, tratado em [13] através do uso de distâncias espectrais e em [14] usando modelos hierárquicos para avaliar casamentos entre segmentos e, por conseguinte, a associação geral, tratando-se, assim, de um procedimento de composição da associação de segmentos. Fazendo menção ao problema de reconhecimento de padrões em pontos, nos conduzimos ao trabalho apresentado em [15], o qual mostra o problema de *Shape Matching* sendo avaliado como um problema dessa natureza. *Shape Matching* é um problema de reconhecimento que pode ser resolvido através da associação de pontos com a transformação das características de contorno de duas imagens em dois conjuntos de pontos, onde se procede com a devida associação.

Em [3] o autor cita trabalhos pioneiros em que o problema de reconhecimento de padrões de pontos é tratado como um problema de associação entre grafos sendo citado o trabalho de [16] onde é destacado que o problema de correspondência de padrões de pontos se trata de um dos mais importantes problemas inseridos no campo de Reconhecimento de Padrões.

1.1 Associação de formas e o reconhecimento de padrões de pontos

O reconhecimento de determinado padrão em uma imagem é efetivado através de mapeamentos entre imagens de entrada (objeto de busca) e imagens de modelo (objetos de comparação). Na abordagem adotada neste trabalho, as imagens são representadas computacionalmente por grafos e o mapeamento entre eles, o casamento desejável entre as imagens comparadas.

Na representação por grafos, vértices são associados aos atributos da imagem e arestas, às relações espaciais entre estes atributos.

As operações aplicadas aos grafos devem ser tais que retornem um conjunto de associações representativas do melhor casamento entre eles, através da associação de seus conjuntos de vértices.

A Figura 1 traz um exemplo do estabelecimento de associações entre dois conjuntos de pontos que traduz o reconhecimento de padrões de pontos. Conforme descreve [16], na Figura 1 temos um grupo a ser identificado (pontos brancos) e uma base de dados para pes-

quisa (pontos pretos) e a relação de associação entre cada ponto representada pelas linhas tracejadas.

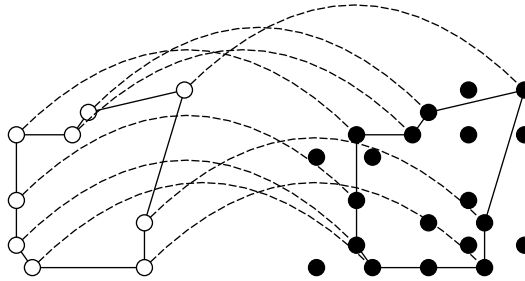


Figura 1: Dois conjuntos de pontos e as relações entre eles

Em [3] destaca-se o uso da forma (*shape*) como fundamental para a pesquisa em bancos de dados de imagens. A Figura 2 exemplifica o processo de reconhecimento de um objeto por associação de pontos, através de sua forma.

A Figura 2(a) mostra a imagem de entrada a ser pesquisada. Na Figura 2(b) temos uma imagem do banco de dados (protótipo). Os grafos que representam cada imagem, de entrada e do banco, respectivamente, são apresentados na Figura 2(c) e na Figura 2(d) é apresentado o casamento obtido pela associação entre os pontos com alguns pontos não associados (o que é permitido dada a natureza de nosso trabalho).

Dentre os desafios encontrados na tarefa de caracterização do casamento entre dois conjuntos de pontos está a correta classificação das relações entre esses conjuntos uma vez que tal trabalho possui limitações principalmente no que se refere às diferenças de tamanhos entre as imagens comparadas (e os conjuntos de pontos obtidos) e também no esforço computacional despendido pelas técnicas empregadas visando obter a associação entre os grafos (que são as estruturas de representação computacional). A descrição do Problema de Associação de Pontos é detalhada na Seção 3.4.

Este problema é um problema NP-completo [3], o que justifica a utilização de técnicas de otimização em algoritmos conhecidos para casamento de padrões, através da utilização de abordagens utilizadas em outros campos de pesquisa dentro da área de computação.

1.2 Objetivos

Esse trabalho visa estudar o problema de reconhecimento de padrões de pontos (PRPP) aplicado na identificação de associação de formas obtidas por imagens e aperfeiçoar méto-

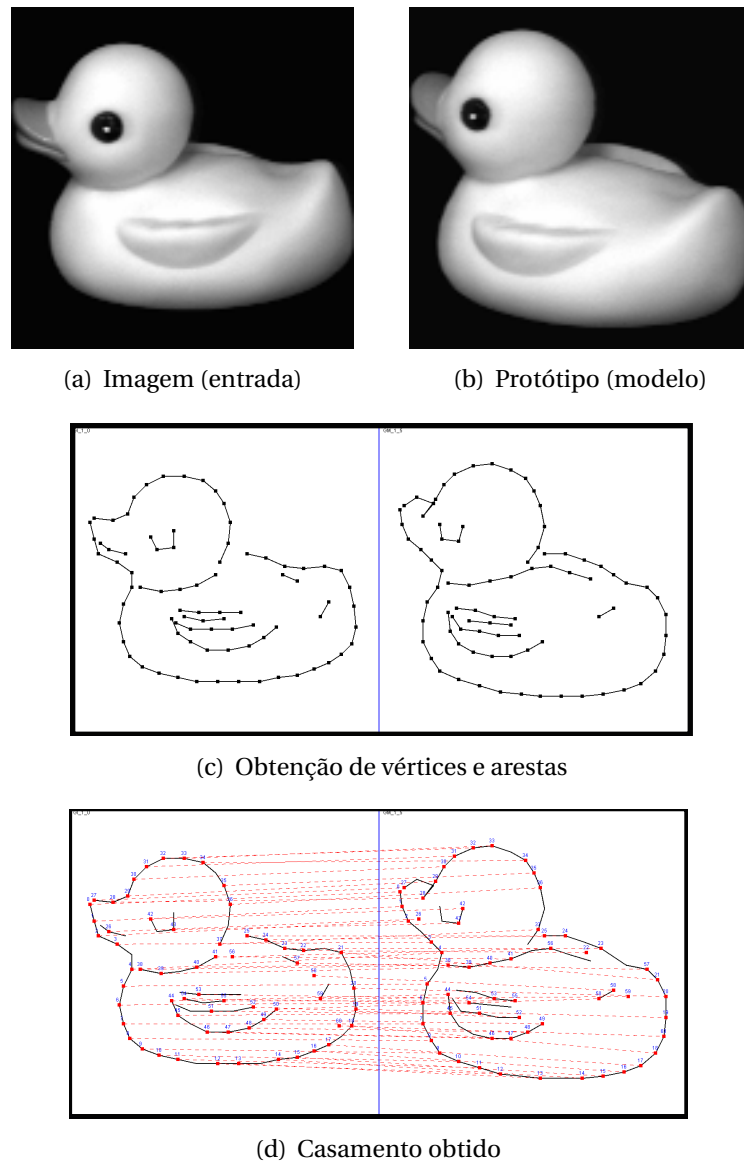


Figura 2: Processo de associação

dos de classificação bem como empregar conhecimentos heurísticos a fim de obter melhorias nos resultados de reconhecimento já conhecidos na literatura.

O processo deste trabalho dará conta de:

1. Elaborar um classificador KNN, que usa a regra dos K vizinhos mais próximos baseado no classificador descrito em [1] e [3], para averiguar a melhor associação entre conjuntos de pontos;
2. Aplicar um método de avaliação multicritério, conforme as ideias apresentadas em [17] e [18], que visa auxiliar na tomada de decisão do custo da associação de arestas (termo quadrático da função objetivo proposta em [3]);

3. Implementar um método de melhoria baseado na heurística 2opt [19], atentando para os movimentos de troca de posição descritos em [20].

1.3 Organização do trabalho

Este trabalho será formulado conforme a seguinte estrutura: no capítulo 2 será apresentada a revisão bibliográfica que diz respeito ao tratamento de imagens para as tarefas de segmentação e reconhecimento de imagens, bem como o trato no reconhecimento de padrões através da correspondência de pontos; no capítulo 3 detalha-se o modelo utilizado para o reconhecimento de formas em imagens e a formulação atribuída; no capítulo 4 são apresentados os métodos computacionais, as funções de avaliação que custeiam as associações e os algoritmos implementados para as abordagens propostas além do método de otimização utilizado e detalhes característicos para sua implementação.

O capítulo 5 mostra os resultados obtidos pelos testes computacionais aos quais submeteu-se a técnica aqui elaborada e, por fim, no capítulo 6 são expostas considerações sobre os estudos efetuados, as conclusões obtidas neste trabalho e propostas futuras.

2 *Revisão Bibliográfica*

Neste capítulo apresentamos uma revisão bibliográfica a respeito:

- Da tarefa de reconhecer um padrão;
- Da empregabilidade dessa tarefa aplicada à correspondência de grafos, onde estes podem ser aplicados como representação para o tratamento de imagens (segmentação e reconhecimento);
- Do reconhecimento de formas através do casamento de pontos (dada a concepção onde assumimos estes como sendo vértices de um grafo).

2.1 O reconhecimento de padrões

Na definição de [21] citada por [22] um padrão é uma entidade, vagamente definida, a qual poderá ser nomeada, sendo qualquer entidade que desperte algum interesse em se reconhecer e/ou identificar. Em outras palavras, um padrão é qualquer coisa que possa ser classificado tal como uma série de automóveis que podem ser do tipo hatch ou sedan dadas as suas características, por exemplo.

Segundo [23] o reconhecimento de padrões é a parte da ciência que infere a respeito da classificação de objetos em categorias ou classes e, dependendo da aplicação desejada, tais objetos podem ser imagens, sinais de onda ou qualquer tipo de medida que necessite de uma classificação.

Os autores de [24] destacam que o reconhecimento automatizado, descrição, classificação e o agrupamento de padrões são abordagens importantes em uma variedade de disciplinas científicas, como biologia, medicina, visão computacional, inteligência artificial e sensoriamento remoto.

Áreas em que o reconhecimento de padrões tem papel indispensável, tais como bioinformática (nas aplicações sobre o sequenciamento de cadeias de genoma), *Data mining* (na

busca de padrões nas informações presentes em bancos de dados), reconhecimento biométrico entre muitas outras são citadas em [1].

Basicamente existem dois tipos de problemas relacionados ao reconhecimento de padrões, o reconhecimento supervisionado e o não supervisionado. No não supervisionado, o problema consiste em descobrir a estrutura do conjunto de dados, ou seja, se há grupos dentro do conjunto de dados e quais as características que fazem os objetos desse grupo serem semelhantes entre si. No reconhecimento supervisionado cada objeto no conjunto de dados vem com um rótulo de classe pré-designado. Dessa forma, o que se tem a fazer é treinar um classificador para rotular os objetos sem classificação. Na maioria das vezes o processo de rotulação não pode ser descrito de uma forma algorítmica, daí, faz-se uso de um processo de aprendizagem computacional para apresentar rótulos para esse objeto ainda não classificado [25].

A Figura 3, baseada nos trabalhos de [25] e [26], mostra um fluxograma de como esses processos podem ser executados.

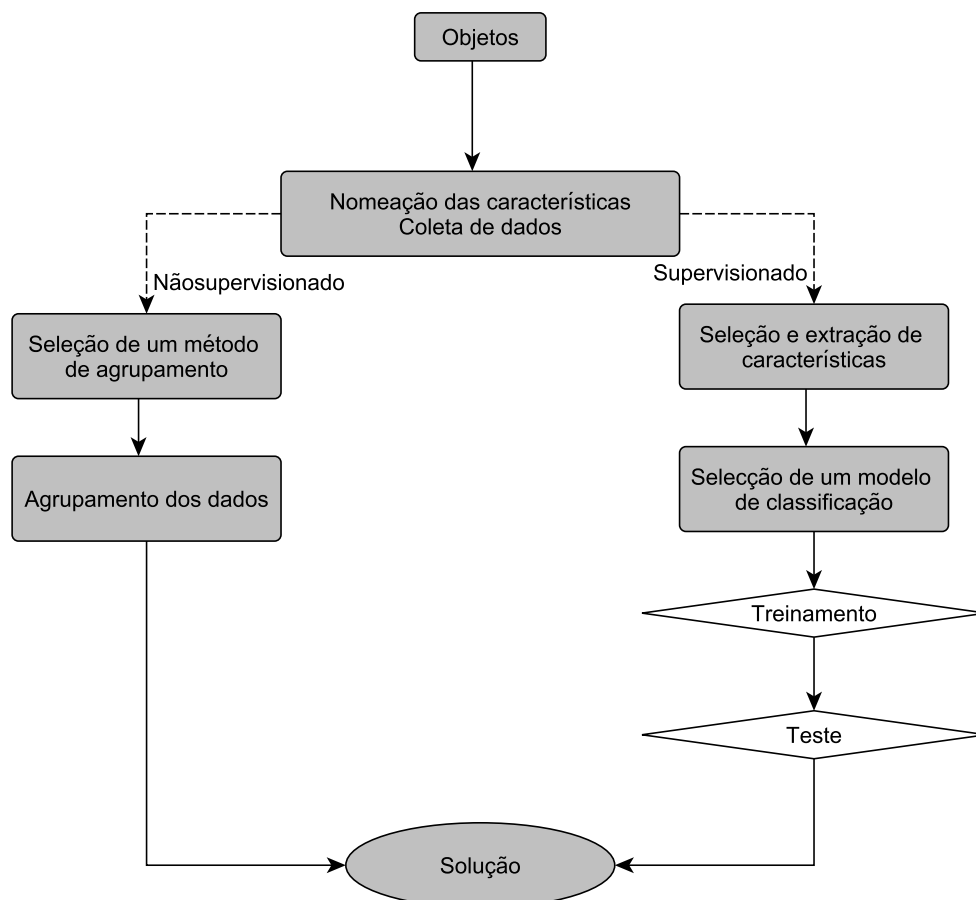


Figura 3: Fluxo de trabalho para reconhecimento de padrões

Muitos trabalhos tratam do reconhecimento de padrões em imagens o qual, como mencionado anteriormente, tem sua motivação baseada no uso de imagens nos mais diversos setores de pesquisa, desde a medicina, onde se utilizam imagens buscando diagnosticar doenças ou planejar novos tratamentos ([27]), na biologia onde se atém o fato do reconhecimento facial como em [28] até o uso em determinação de localizações geográficas ([29]).

Segundo [30], determinar a localização de objetos em uma imagem, sua topologia e sua estrutura são aspectos estudados dentro da área de tratamento de imagens através do uso de computação e damos especial importância a este ramo do reconhecimento de padrões, com representação em grafos, por ser o foco de nosso trabalho.

2.1.1 O processo de reconhecimento de um padrão

Em um problema de reconhecimento de padrões sobre um conjunto de treinamento de vetores de características rotuladas, o objetivo é encontrar um mapeamento que classifique os dados de entrada corretamente [31]. Em [31] e [32] é descrito como se procede o reconhecimento de um padrão, o qual deve ser concebido estabelecendo um mapeamento entre o conjunto de informações desconhecidas e um conjunto de dados pré-conhecidos.

A figura 4 mostra um sistema genérico de reconhecimento de padrões.

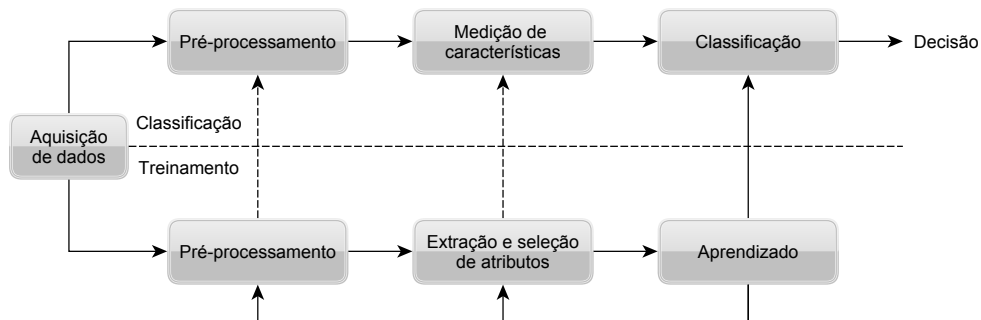


Figura 4: Sistema de reconhecimento de padrões. Fonte: [1]

Segundo [33], existem paradigmas específicos para proceder com o reconhecimento de um padrão e entre esses paradigmas, o mais difundido é o paradigma estatístico. Não obstante, outro paradigma denominado paradigma estrutural, que se vale principalmente do uso de estruturas de representação como grafos, é citado como uma importante abordagem quando informações estruturais são dados importantes para a caracterização de um padrão.

- Paradigma estatístico

Nesse paradigma, como descrito em [23] apud. [1], a partir de um conjunto c de classes designadas como w_1, w_2, \dots, w_c e, dado um padrão desconhecido x , um sistema reconhecedor de padrões é tal que, com o auxílio de um pré-processamento e a extração e seleção de características, é capaz de associar o padrão x à uma classe w_i rotulada por i .

A grande dificuldade ao se abordar o reconhecimento de padrões usando o paradigma estatístico, conforme menciona [33], está na aquisição das características que compõem as hipóteses estatísticas.

- Paradigma estrutural

Para esse paradigma, segundo [26], o que se propõe é dar ênfase na descrição da estrutura que compõe o padrão, ou seja, explicar como sub-padrões podem ser analisados de maneira a caracterizar como um padrão é composto, ou seja, esse paradigma leva em consideração a relação de associação para cada parte do objeto que se deseja reconhecer.

O autor de [26] ainda ressalta que esse paradigma é associado com a classificação estatística para poder lidar com problemas mais complexos como o reconhecimento de objetos multidimensionais. Essa abordagem será mais amplamente discutida na seção 2.3.

Outras abordagens também são citadas como a abordagem por agrupamento de dados e redes neurais [26, 33], contudo, detalhes a respeito das mesmas não é nosso foco.

Nas subseções seguintes detalhamos as fases do processo de reconhecimento de um padrão, a saber: o pré-processamento, a extração e seleção de características e classificação.

2.1.1.1 Pré-processamento

O papel dessa fase é trazer para um nível mais alto informações de relevante interesse para o padrão que se deseja encontrar, ou seja, filtrar de forma conveniente informações que sejam necessárias para proceder com o reconhecimento do padrão.

Um exemplo dessa elevação de nível é a detecção de bordas em imagens para averiguar o contorno de objetos [34], dessa forma as demais informações pertinentes à imagem são descartadas por não serem necessárias.

Outro exemplo é a redução de ruídos para eliminar características disformes ao contexto do padrão, como em [3] onde, para sua aplicação de segmentação interativa de imagens, foi reduzido significativamente o número de regiões *watershed*, que se baseia na existência de regiões topográficas em uma imagem a partir da divisão da mesma através da avaliação da

concentração de pixels e dos níveis de cinza desses pixels em subregiões da mesma imagem (mais detalhes podem ser encontrados em [35]).

A figura 5 exemplifica essa fase. Em 5(a) mostramos uma imagem de um pato de brinquedo e em 5(b) mostramos o resultado da elevação de nível de informações onde são preservadas apenas as características do contorno da imagem original e as demais informações são descartadas.

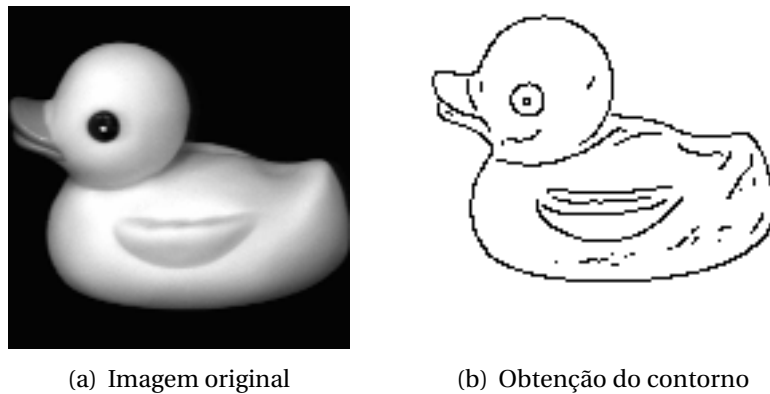


Figura 5: Exemplo de pré-processamento

2.1.1.2 Extração de características

Segundo [32], características devem ser de fácil computabilidade permitindo uma boa visualização das mesmas. Informações geométricas ou estruturais são tipos de recursos avaliados como características. Muitos algoritmos de seleção de características envolvem uma procura de tamanho combinatorial dado o espaço de busca. Esta etapa é de relevante importância pois nela são geradas as informações que serão usadas na computação das funções que indicam custos de associações que determinam o mapeamento entre os grafos comparados, por exemplo.

Um exemplo é dado na Figura 6 onde, das informações pré-processadas, são retiradas características que formam um conjunto de informações que pode ser usado para se obter uma classificação. Na Figura 6(a) temos o contorno de uma imagem e a partir desse contorno são processadas informações que o subdividem em vértices e entre cada vértice pertencente a uma mesma parte do contorno original é atribuída uma aresta de ligação, promovendo a caracterização das informações do contorno através de um grafo como mostrado na Figura 6.

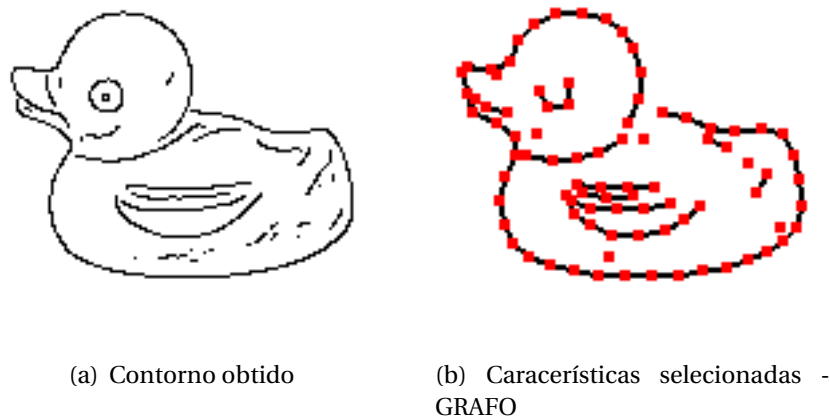


Figura 6: Extração de características - Exemplo

2.1.1.3 Classificação

Segundo [1], tendo um padrão x , desconhecido, que esteja inserido no conjunto X de padrões para testes pertencente a um espaço de características, e sendo o conjunto Ω que engloba todas as classes de padrões existentes, um classificador é uma função $\Upsilon : X \rightarrow \Omega$, tal que $\Upsilon(x) = w_i$, sendo w_i o i -ésimo padrão pertencente a Ω . Dessa forma, um classificador é uma função que recebe um padrão desconhecido como parâmetro de entrada e retorna como resposta uma classe conhecida a qual o padrão de entrada deve pertencer. Os objetivos essenciais do trabalho com um método de classificação são minimizar os erros de classificação bem como permitir uma classificação computacionalmente eficiente.

Usaremos classificadores baseados na regra de classificação dos K vizinhos mais próximos - KNN (do inglês *k-Nearest Neighbor*) por ser essa a estratégia de classificação adotada em nosso trabalho. Mais detalhes sobre classificadores podem ser encontrados em [2].

Regra dos K vizinhos mais próximos

Essa regra de classificação, como descreve [1], é independente de processamento na fase de treinamento, não sendo necessário prover as distribuições de probabilidades das classes encontradas no conjunto de classes existentes. Não obstante, a mesma é dependente da existência de um grande número de padrões de treinamento (aqueles que se conhece a classificação) para que se possa tratar uma estimativa sobre a qual padrão o objeto sem classificação deve ser atribuído e proceder com a classificação, propriamente dita, em uma única tarefa.

De acordo com [2], essa regra atribui a um dado padrão x , inicialmente desconhecido, a classificação quanto ao rótulo de classe que mais vezes aparece entre as K amostras mais

próximas (com menor custo de associação) no conjunto de dados de treinamento.

Essa regra de classificação, segundo [1], pode ser computada através das seguintes fases:

1. Cálculo dos custos de associação entre o padrão desconhecido x com os padrões do conjunto de treinamento;
2. Verificação das classes a que pertencem os K padrões de menor custo (ou mais próximos);
3. Associação à classe cujo rótulo mais frequentemente aparece entre os K padrões de menor custo.

Um caso especial dessa regra é dado quando temos $K = 1$, assim teremos o caso do classificador de vizinho mais próximo (1NN). Para um conjunto de treinamento grande e com classificações elaboradas de maneira errada, deve-se dar preferência ao classificador KNN [1], dado que, com a comparação entre as melhores classificações pode eliminar uma classificação errada que esteja na primeira posição entre as melhores classificações.

Um exemplo visual baseado em [2] é mostrado na Figura 7 onde, dado um padrão x , avalia-se através da distância euclidiana quais os pontos mais próximos a este padrão e, a partir dessas distâncias, o mesmo é classificado como sendo um ponto branco (círculo vazado) dada a proximidade com os padrões ao seu redor. Este exemplo usa como base um classificador onde $K = 5$, ou seja, o padrão x é classificado como pertencente à classe que é mais frequente entre os cinco pontos mais próximos a ele.

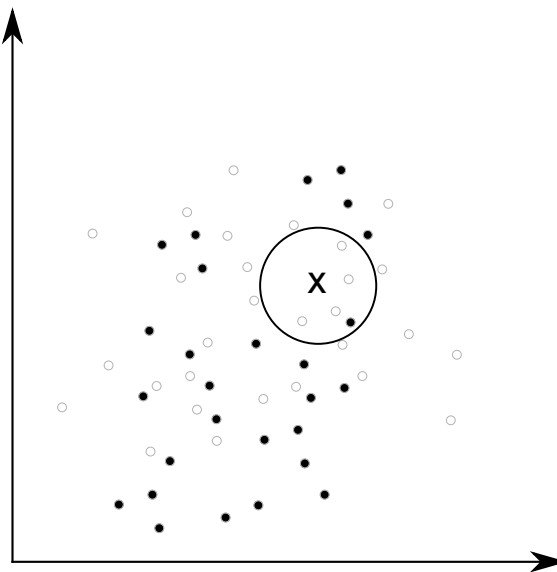


Figura 7: Regra KNN. Fonte: [2]

2.2 Problemas abordados em imagens

Segundo [36], um sistema de visão artificial emprega fases distintas de tratamento sobre imagens para que se possa proceder com o trabalho computacional a que se destinam tais imagens. Essas fases geralmente são divididas em: aquisição, pré-processamento, segmentação, extração de características e reconhecimento ou associação.

As fases de segmentação e associação estão presentes em inúmeros trabalhos encontrados na literatura sendo abordados como problemas em aberto inseridos na área da visão computacional. Devido a essa grande presença na literatura, esses problemas serão abordados nesta seção a fim de dar respaldo ao trabalho desempenhado mostrando as dificuldades encontradas e motivações para tentar resolvê-los.

2.2.1 Segmentação de imagens

Uma importante área dentro do reconhecimento de padrões em imagens onde é possível o emprego de grafos como representação computacional é a que trata da segmentação de imagens. Essa área tem como base a extração de informações em uma imagem, dividindo-a, e transformando tais informações em grafos, os quais são utilizados em tarefas posteriores no reconhecimento de padrões.

A tarefa de segmentação de imagens, conforme [37], consiste em classificar uma imagem em regiões homogêneas distintas baseando-se em uma seleção de propriedades através de objetos únicos, como centróides, intensidade de cor, coordenadas espaciais, textura, formas etc. com o intuito de garantir uma melhor descrição e compreensão das informações presentes na imagem.

Reproduzimos aqui a definição formal para o problema de segmentação de imagens mostrada em [37]: dada uma imagem que possui uma permutação específica de valores de intensidade p_{ij} , considerando $P = \{p_{ij}, i \leq m, j \leq n\}$ o conjunto de valores de intensidade em uma imagem $m \times n$ e K o número de segmentos/classes dessa imagem. Uma função c pode ser definida como $c : P \rightarrow [0 \dots K]$ para descrever o processo de segmentação de uma imagem. Uma classificação para tal imagem baseada em 256 níveis de cinza implica no agrupamento em termos de níveis de cinza $\{t_k, 0 \leq k \leq K\}$, onde t_k representa cada agrupamento, tal que, $0 = t_0 \leq t_1 \leq \dots \leq t_{k-1} \leq t_k = 255$. Todos os pixels com níveis de cinza em $[t_{k-1}, t_k) = s_k$ pertencem à classe k . Com isso, o objetivo da segmentação é estabelecer uma escolha ideal $S_I^* = \{s_k^*, 1 \leq k \leq K\}$, onde $S_I = \{s_k, 1 \leq k \leq K\}$ indica uma classe da imagem I . Essa escolha ideal é um conjunto de pixels e sua escolha dada como “ideal” deriva de cada

aplicação, por exemplo, para uma aplicação que visa agrupar apenas dois conjuntos, um com pixels que possuam nível de cinza igual a 0 e outro conjunto com os demais pixels.

Em [3] o autor infere que, basicamente, o objetivo da segmentação de uma imagem é dividi-la em diferentes regiões. Os propósitos dessa divisão, em geral, englobam a extração de objetos e/ou sobreposição do fundo da imagem para enfatizar características pré-selecionadas da imagem. Exemplos deste tipo de tarefa podem ser encontrados em [38, 39, 40].

Em [41] é apresentada uma técnica para segmentar uma imagem com base em [42] onde o usuário participa do processo de segmentação marcando alguns pixels da imagem que farão parte da região de interesse e os outros pixels que não fazem parte dessa região, fazem parte do fundo da imagem segmentada. Esse tipo de segmentação é denotado em [38] por segmentação semi-automática. Dessa forma, o objetivo da abordagem semi-automática é encontrar a melhor segmentação que satisfaça as limitações impostas pelo usuário. Após essa marcação feita pelo usuário, a imagem é então processada, fazendo uso da segmentação em sua forma mais simples a qual se vale da separação de áreas com pixels que possuam níveis de cinza coincidentes, e as zonas de segmentação finais, após a avaliação de pixels conforme a delimitação do usuário, são demarcadas.

Em [43] um procedimento semi-automático de segmentação foi apresentado de tal forma que o usuário seleciona áreas de seu interesse presentes na imagem (através de traços desenhados na própria imagem) e essas áreas são rotuladas e unificadas formando regiões que são apresentadas como resposta.

Vários trabalhos da literatura tratam do problema de segmentação semi-automática, não obstante, há trabalhos cuja abordagem é a segmentação automática, quando não há interação do usuário. Em [44] temos um exemplo desse tipo de abordagem onde os autores trabalham com o particionamento de uma imagem através do grupos de cores presentes.

Em [45], uma etapa inicial baseada em supersegmentação da imagem através da transformada *watershed* [35, 46] é efetivada para, posteriormente, fundir iterativamente segmentos que se assemelham de acordo com critérios estabelecidos assim como níveis de cinza, por exemplo. Mais detalhes a respeito da segmentação automática podem ser encontrados em [44, 47, 48].

A segmentação automática também é abordada em [49], onde é vinculada a um método que trata a imagem através da observação de suas cores, usadas como atributos para as regiões a serem segmentadas. Vale observar que os autores enfatizam a utilização de grafos

como um potente artifício para o tratamento e codificação de informações bem como para a representação da topologia da imagem. Naquele trabalho, os grafos indicam as relações de adjacência para cada parte segmentada da imagem.

Em geral o processo de segmentação leva em consideração algumas características básicas da imagem, como tonalidade de cada pixel que a forma. Um esquema de segmentação simples, com a utilização de uma etapa de supersegmentação inicial, baseado no trabalho de [43] e [50] é mostrado na figura 8.

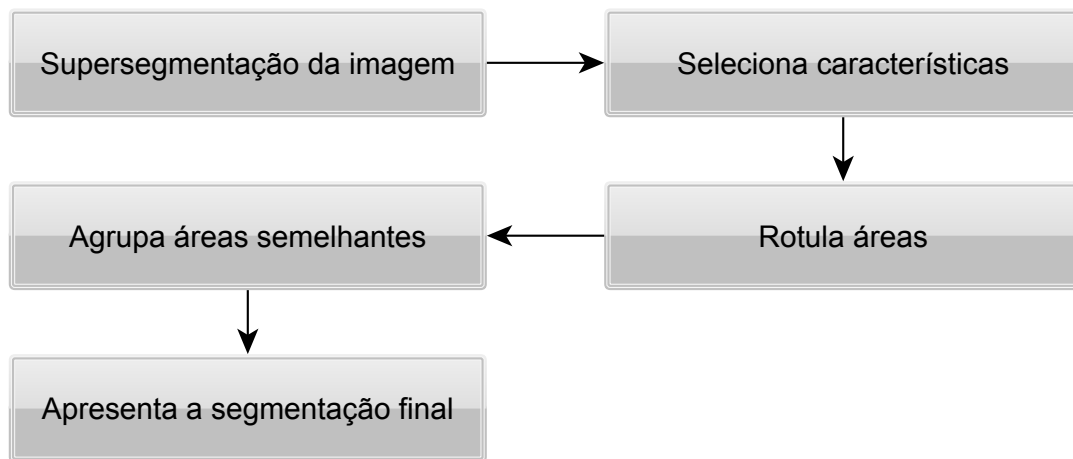


Figura 8: Procedimento básico de segmentação de uma imagem

2.2.2 Correspondência de imagens

A correspondência entre imagens é a parte da visão computacional incumbida de reconhecer uma imagem, ou parte de uma imagem, através de sua comparação com outra imagem previamente classificada dentro de uma base de conhecimento.

O problema da correspondência de imagens está em estabelecer a correta relação entre as duas imagens a fim de que o reconhecimento proceda de maneira satisfatória. Esse reconhecimento se baseia na comparação das características de cada imagem e a grande dificuldade encontrada nessa comparação está na definição dos dados a serem comparados e como os mesmos serão comparados.

Em [51] encontramos um exemplo desse tipo de tarefa. Naquele trabalho o que se deseja é estabelecer a associação entre duas imagens de eletroforese bidimensional que é um método de separação de proteínas comumente usado na área de biociência no campo da pesquisa proteômica, conforme [52]. Nessa aplicação, a imagem é dada como um conjunto de

pontos (ou manchas) os quais representam acúmulos de proteínas, dessa forma duas imagens devem ser associadas (Figura 9) visando obter um estudo sobre as mudanças protéicas que as distinguem, ou seja, indicar grupos proteicos semelhantes.

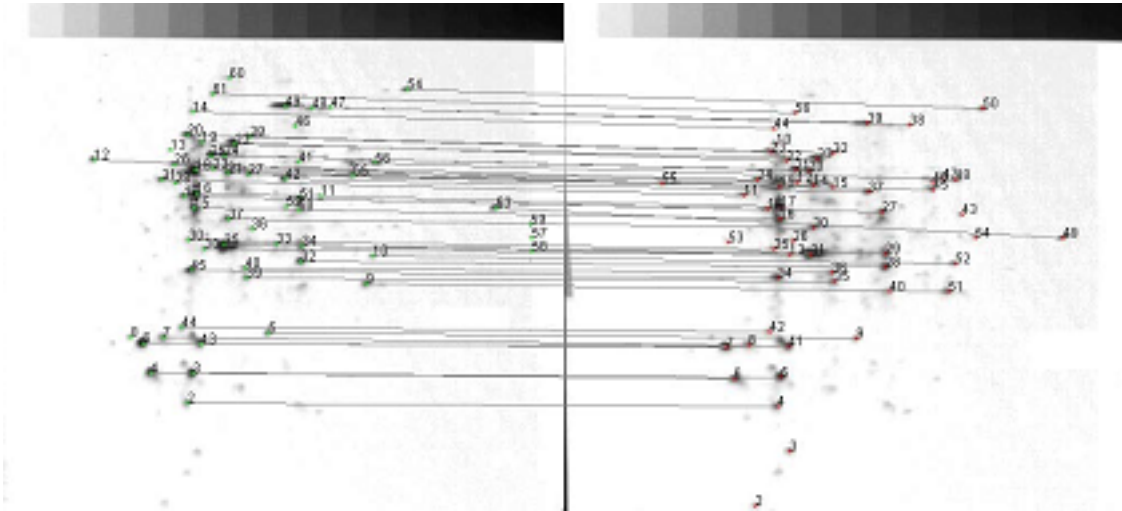


Figura 9: Associação de imagens que representam grupos de proteínas. Fonte: [3]

Outros trabalhos abordam a associação de imagens de diferentes tamanhos. Nesses casos, geralmente, o objetivo é encontrar uma imagem de menor tamanho inserida na imagem maior através da aquisição e comparação de características presentes nas duas imagens. Um exemplo desse tipo de abordagem é apresentado em [29] onde imagens devem ser associadas a parte de uma imagem maior. No caso desse trabalho em especial, o que se tem são trechos viários associados a um mapa mais abrangente e a Figura 10 exemplifica as associações e o tipo de associações almejadas.

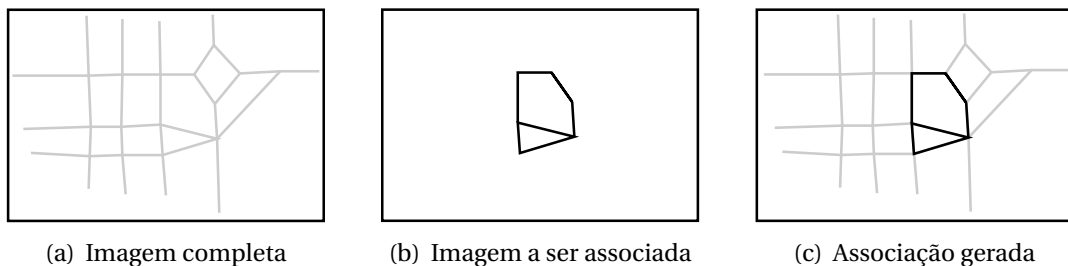


Figura 10: Associação de imagens - Exemplo

Um ramo importante da visão computacional é o problema de detecção de objetos em imagens. Essa tarefa pode ser entendida como a tarefa de reconhecer uma imagem em uma outra imagem, por exemplo, dada uma imagem de um liquidificador, desejamos que o mesmo seja identificado na foto de uma cozinha. No trabalho de [53], onde utilizam um procedimento baseado na extração de informações de pontos da imagem agregando dados

estruturais como forma de aperfeiçoar o processo de identificação, os autores argumentam que o trabalho de detecção de objetos pode ser encarado como a tarefa de decidir se uma determinada imagem, que representa um objeto, está inserida em outra imagem, como uma cena de filme, além de ser capaz de determinar a sua posição na referida cena.

Outra aplicação para correspondência de imagens é tal que tenta averiguar a semelhança entre objetos diferentes para avaliar se pertencem à mesma classe de objetos. Um exemplo desse tipo de abordagem dado em [54] é a comparação de imagens de dígitos manuscritos, os quais são comparados pela sua estrutura de contorno.

A comparação de imagens através de seu contorno é um tipo de problema abordado em associação de imagens denominado de Casamento de Formas ou *Shape Matching*. A Figura 11 mostra duas imagens, provenientes do banco de dados MNIST¹, que pertencem à mesma classe. Neste caso a classe indica que são referenciados como dígitos que representam o número 2 (dois). Mais detalhes a respeito desse banco de imagens serão apresentados no Capítulo 5 onde são discutidas as bases de imagens usadas nos testes computacionais.



Figura 11: Exemplos de dígitos manuscritos, ambos representando o número dois

Os problemas mencionados nessa seção são amplamente abordados na literatura e possuem formas diferentes de serem tratados. Nós daremos ênfase para as abordagens relacionadas a imagens utilizando grafos como representação, mais especificamente para a tarefa de associação entre imagens.

2.3 Correspondência de Imagens via correspondência de grafos

A tarefa de relacionar uma imagem a outra não é trivial em termos computacionais uma vez que se faz necessário adquirir informações, processá-las e, por fim, classificá-las.

Uma forma utilizada para modelar as características contidas em imagens se dá pela utilização de grafos, uma vez que a estruturação de uma imagem como um grafo gera um mecanismo de visualização simples a respeito das informações pertinentes à imagem.

¹<http://yann.lecun.com/exdb/mnist/>

Com a utilização de grafos para representar imagens computacionalmente, a tarefa de estabelecer uma correspondência entre imagens passa a ser a tarefa de estabelecer uma correspondência entre grafos e essa é grande vantagem dessa abordagem, uma vez que o estudo sobre grafos é muito mais amplo possuindo bastante material de pesquisa.

2.3.1 Representação de uma imagem como um grafo

Se aproveitar do uso de algoritmos já bastante difundidos na teoria dos grafos e ainda se valer de uma representação compacta das informações presentes nas imagens estimula a utilização de um grafo, aproveitando seu potencial para a representação e classificação de padrões. O paradigma estrutural de reconhecimento de padrões, por exemplo, pressupõe a relevância de se ater a informações estruturais quantificadas em descritores relacionais (grafos).

Em geral, o tipo de grafo empregado nos trabalhos estudados estão como grafos de atributos relacionais (ARG). Nesta seção apresentamos conceitos básicos relativos à estrutura matemática geral de grafo e de grafo relacional, além das particularidades da generalização de um grafo relacional para um grafo de atributos relacionais, suas representações e a indicação de trabalhos onde houve o emprego de tais estruturas.

2.3.1.1 Definições básicas sobre grafos

Em [55] é apresentada uma série de definições sobre grafos, as quais resumimos nessa subseção:

- **Grafo:** é uma estrutura $G = (V, E)$ tal que V representa o conjunto (finito) de vértices pertencentes ao grafo e $E \subseteq V \times V$ representa o conjunto de arestas do grafo. As cardinalidades de V e E são expressas por $|V|$ e $|E|$, respectivamente. Diz-se arestas para o caso de serem pares não ordenados e para o presente trabalho as ligações entre vértices de um grafo serão simplesmente denominadas de arestas.
- **Grafo Relacional:** é um grafo valorado, ou seja, possui pesos associados às suas arestas, ou seja, dado o grafo $G = (V, E)$ a cada aresta $e = (v_a, v_b) \in E$ temos um peso p_{ab} associado às mesmas.

2.3.1.2 Grafos de atributos relacionais (ARG)

Daremos especial atenção para os grafos de atributos relacionais ou ARG (proveniente do inglês *Attributed Relational Graph*), por ser uma abordagem bastante aplicada em diversos problemas que envolvem o reconhecimento de padrões.

A definição de um ARG pode ser encontrada em [33] e [55]. A estrutura ARG se trata de uma extensão de Grafo Relacional em que são agregadas informações a respeito das características estruturais do padrão, bem como características simbólicas (cor por exemplo).

A definição formal encontrada em [33, 55] nos diz que um ARG é uma 4-tupla $G = (V, E, \mu, \omega)$ onde V indica o conjunto de vértices e E representa o conjunto de arestas do grafo e cada vértice possui um vetor de atributos dado por $\mu: V \rightarrow L_V$. Analogamente, existe um vetor de atributos associado a cada aresta do grafo definido por $\omega: E \rightarrow L_E$. Exemplos da utilização de ARGs podem ser encontrados em [56], [57] e [58].

Muitas aplicações em visão computacional utilizam imagens como objetos de pesquisa e as formas de trabalho adotadas para cada aplicação podem ser diferenciadas de acordo com a abordagem sugerida. Um exemplo é uma aplicação que visa responder ao questionamento sobre a existência ou não de determinado objeto em uma cena de vídeo (a qual pode ser encarada como uma imagem).

Algumas formas de abordar os problemas encontrados em correspondência de imagens são mais frequentemente relatadas na literatura merecendo especial atenção sobre a condução do processo de associação entre imagens.

Na próxima seção discutiremos algumas dessas abordagens aplicadas ao problema de associação de imagens.

2.3.2 Formas de abordagem para resolução de problemas de associação em imagens

A tarefa de reconhecimento de imagens implica em várias formas de associação entre elas, dependendo do interesse da aplicação. Apresentaremos algumas delas nesta seção.

As imagens exibidas nesta seção, usadas como exemplos das abordagens mencionadas, são provenientes do banco de imagens MPEG-7² CE-Shape-1[59]. Esse banco de imagens é formado por 1400 imagens com 70 classes de 20 imagens cada, sendo composto por silhuetas de vários objetos.

²<http://mpeg.chiariglione.org/standards/mpeg-7/mpeg-7.htm>

2.3.2.1 Comparação de uma imagem com parte de outra imagem

Em geral, esse tipo de abordagem é encontrado em situações em que se procura determinado objeto em uma imagem [53]. Essas abordagens inferem geralmente sobre o seguinte questionamento: a imagem a existe “dentro” da imagem A . Como exemplo, na Figura 12 gostaríamos de responder se a imagem mostrada na Figura 12(a) está presente ou não na imagem da Figura 12(b). O esperado, nesse exemplo, é uma resposta afirmativa. A Figura 13 mostra como seria a representação da resposta esperada para este exemplo. Se as imagens são representadas por grafos, o problema de reconhecimento pode ser representado, com algumas adaptações, por um problema conhecido em Teoria dos grafos: o problema de isomorfismo de subgrafos.

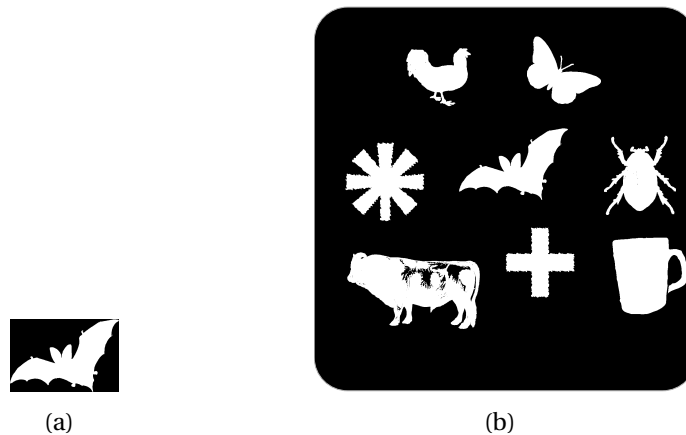


Figura 12: Exemplo de comparação. A imagem 12(a) está na imagem 12(b)?

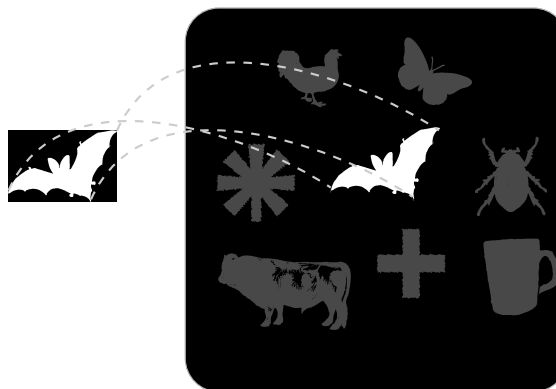


Figura 13: Resposta esperada para a presença ou não de uma imagem em outra

2.3.2.2 Comparação de duas imagens

Com essa abordagem o que se espera é responder, qual o grau de semelhança entre duas imagens, podendo ser total, caso em que as duas imagens são idênticas.

Por exemplo, na Figura 14 a questão é responder com qual das duas imagens (Figuras 14(b) e 14(c)) a imagem presente na Figura 14(a) se assemelha mais.

Esse tipo de tratamento é base para as comparações efetuadas em bancos de imagens e com essa resposta pode-se inferir a respeito de qual classe uma imagem pertence, como será mostrado na próxima seção. Esse tipo de comparação, se as imagens são representadas por grafos, pode ser descrito pelo problema de Isomorfismo de grafos [33].

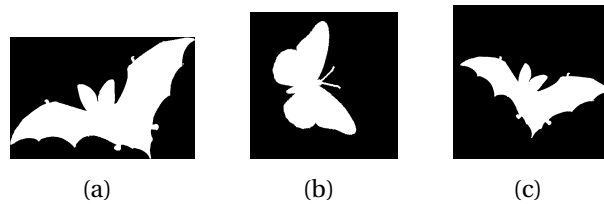


Figura 14: Exemplo de comparação. A imagem 14(a) é mais semelhante a 14(b) ou a 14(c)

2.3.2.3 Comparação de uma imagem com um banco de imagens

Nesse tipo de abordagem, procede-se com várias comparações entre imagens como descrito na seção anterior, ou seja, uma imagem é comparada com outras imagens, uma a uma. É um trabalho que, em geral, é efetuado visando obter uma medida de comparação um-para-muitos e, por conseguinte, a atribuição da imagem sem classificação a uma classe já conhecida.

Exemplos desse tipo de abordagem podem ser encontrados em [6], [60] e [61] onde testes são efetuados sobre silhuetas de KIMIA³, o qual é composto por 99 imagens sendo 9 classes com 11 imagens cada, e também com a base de imagens MPEG-7 [59].

A Figura 15 exemplifica esse tipo de abordagem onde a Figura 15(a), após a comparação com as imagens do banco em 15(b) deve ser classificada como uma imagem de um “osso”.

Os problemas aqui expostos, sendo modelados através de grafos, como é a proposta desse trabalho, podem ser resolvidos através do casamento de grafos. A próxima seção conceitua esse tema mostrando como o mesmo pode ser tratado.

³<http://www.lems.brown.edu/vision/researchAreas/SIID/>

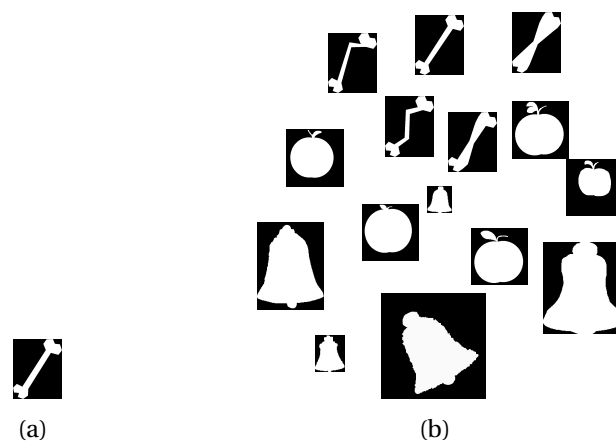


Figura 15: Exemplo de comparação. A imagem 15(a) é comparada com as imagens do banco 15(b)

3 O Problema de Associação de Grafos

Segundo [33], o reconhecimento de um padrão com a utilização de grafos deve ser feito de tal forma que as informações estruturais sejam representadas através da modelagem de um grafo.

Com isso pode-se fazer comparações entre as classes desconhecidas e classes de protótipos previamente conhecidos elaborando classificações com base em medidas de *similaridade* e, a partir de tais medidas, inferir uma correspondência entre o conjunto de vértices de um grafo com o conjunto de vértices de outro grafo bem como, os conjuntos de arestas de cada grafo no par de grafos selecionado para comparação de maneira que as informações estruturais de um grafo sejam identificadas no outro.

As formas de se abordar o casamento entre grafos podem ser divididas em dois grupos, a correspondência exata e a inexata. Essas formas de casamento entre grafos serão descritas a seguir.

3.1 Correspondência exata

Em [55] a correspondência exata entre grafos ou subgrafos é tratada sobre estruturas precisamente descritas e consiste em um mapeamento entre os vértices dos grafos, feita através de uma bijeção, com o intuito de manter a estrutura das arestas. De forma análoga, essa ideia pode ser aplicada em isomorfismo de subgrafos onde a comparação é feita entre um grafo e parte de outro grafo (subgrafo).

Dois grafos $G_a = (V_a, E_a)$ e $G_b = (V_b, E_b)$ são ditos isomorfos, [62], se e somente se $|V_a| = |V_b|$ e $|E_a| = |E_b|$, e existir uma bijeção $f : V_a \rightarrow V_b$ tal que, $(v_{a1}, v_{a2}) \in E_a$ se e somente se $(f(v_{a1}), f(v_{a2})) \in E_b$. Um exemplo de isomorfismo é apresentado na Figura 16.

Conforme [33, 55], no caso de isomorfismo de subgrafo, como o exemplo dado na Figura 17, o problema equivale a dizer que, dado dois grafos, $G_a = (V_a, E_a)$ e $G_b = (V_b, E_b)$, G_a possui um subgrafo G_c isomorfo a G_b se existir um subconjunto $V_c \subseteq V_a$ de vértices de G_a , bem

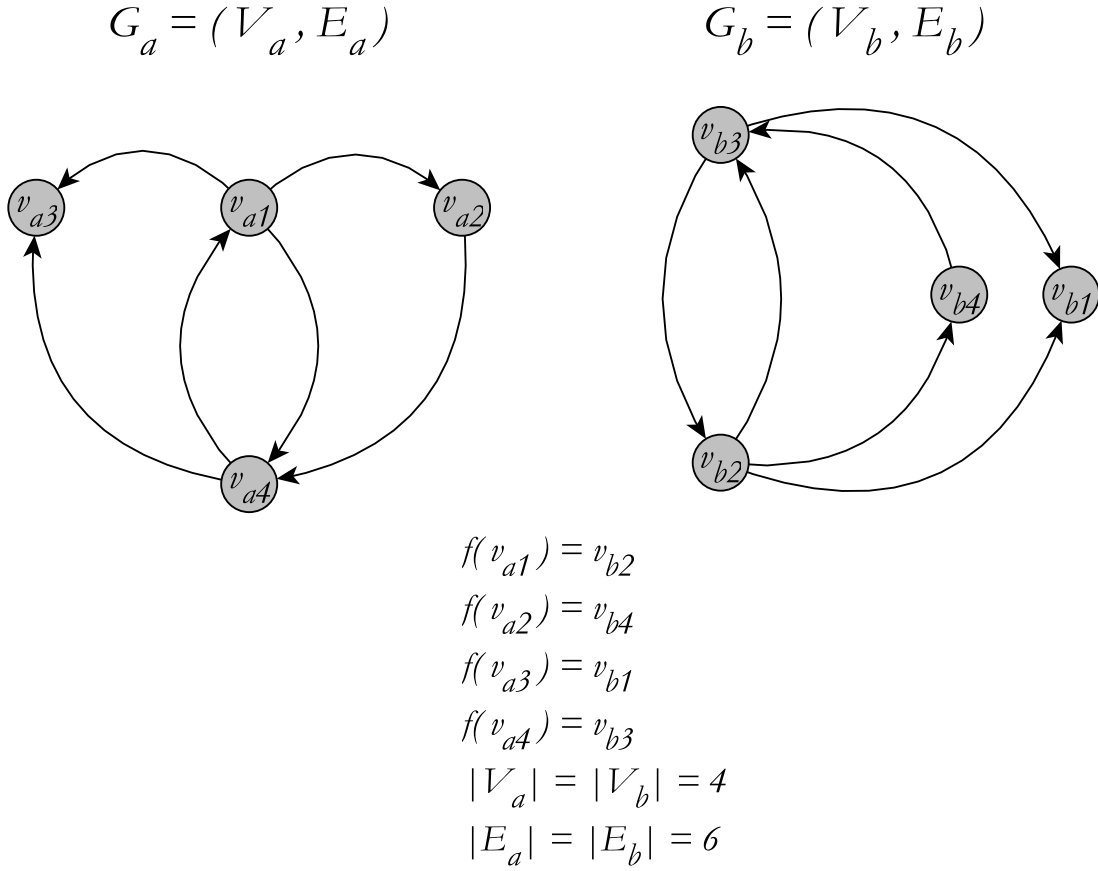


Figura 16: Exemplo de isomorfismo. G_a é isomorfo a G_b

como um subconjunto $E_c \subseteq E_a$ tais que, $|V_c| = |V_b|$ e $|E_c| = |E_b|$ e, ainda, existe uma função biunívoca $f : V_c \rightarrow V_b$ tal que $(v_{c1}, v_{c2}) \in E_c \iff (f(v_{c1}), f(v_{c2})) \in E_b$. Um caso especial e importante encontrado na literatura é o problema do Máximo Subgrafo Comum (MCS, do inglês *Maximum Common Subgraph*), onde se deseja encontrar o subgrafo de maior tamanho pertencente a G_a e que seja isomorfo a G_b [3].

3.2 Correspondência inexata

Segundo [33] grafos são usados na representação de muitas aplicações, mais especificamente se tratando de imagens ([55]), estão propensos a terem imperfeições derivadas do processo de aquisição de informações, variação dos padrões e outros casos de deformações.

Para a tarefa de reconhecimento de imagens representadas por esse tipo de grafos, a abordagem utilizada geralmente é correspondência inexata de grafos onde a associação de vértices que não mantém a condição de associação de arestas é permissível e são associados

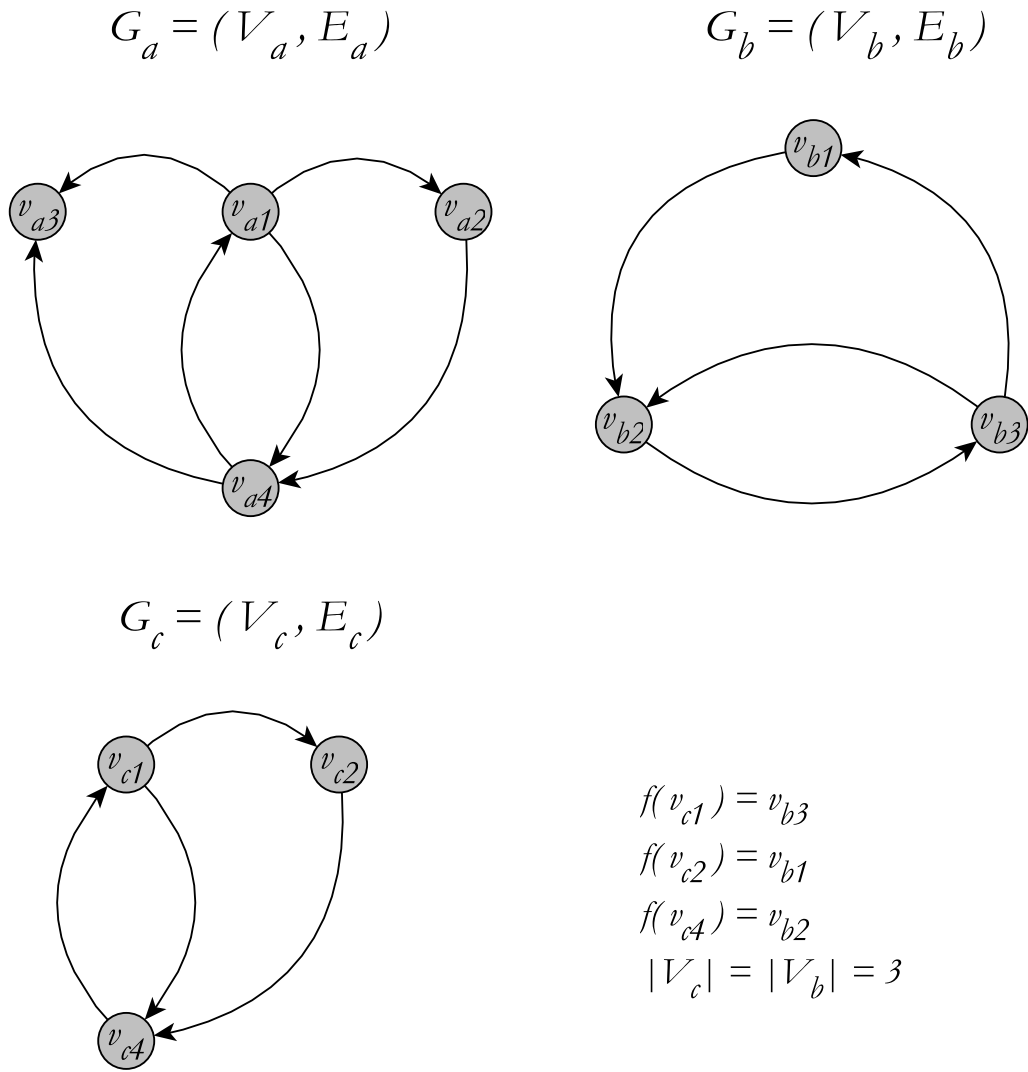


Figura 17: Exemplo de isomorfismo de subgrafo. G_c , subgrafo de G_a , é isomorfo a G_b

custos a cada associação sendo que esses custos são baseados em informações estruturais e aos atributos provenientes dos vértices e arestas. Com isso, o mapeamento, nesse caso, depende da minimização do custo total das associações, sendo que, não há garantia de obtenção da melhor solução para o problema mas sim, de uma aproximação do mapeamento ótimo, como descrito em [33].

Da definição dada em [63] temos que um casamento inexato, ou *matching* inexato, é obtido quando, dados f_1, \dots, f_n todos os possíveis casamentos (inexatos) entre dois grafos G_a e G_b quaisquer, obtemos f^* , o melhor casamento entre G_a e G_b , o que implica, através da função de custo apropriada, dizer que:

$$\text{custo}(f^*) = \min_{f_1, \dots, f_n} \{\text{custo}(f)\}$$

3.3 O reconhecimento de formas

O problema de reconhecimento de formas é encontrado na área de pesquisa sobre reconhecimento de padrões em imagens apresentando características distintas por se tratar basicamente do processo de reconhecer uma imagem através dos detalhes de seu contorno como descrito em [15].

Muitas são as formas de abordar esse problema e em muitos casos se utiliza da comparação entre grafos para proceder com o reconhecimento. Em [4] os autores modelam o problema do reconhecimento de formas como o problema de associação entre árvores, uma vez que traduzem a forma (imagem) através de um *shock graph* que é a árvore extraída pelo esqueleto, ou eixo médio, da imagem adquirido no interior da forma. A extração desse eixo médio é feita pela aquisição do lugar geométrico em que se encontram os pontos interiores da forma que sejam minimamente equidistantes a dois pontos do seu contorno [64]. Esses pontos médios formam o esqueleto da forma avaliando a largura da mesma onde é estabelecido um fluxo que segue das regiões mais estreitas da forma para as regiões mais largas. A Figura 18 mostra um exemplo de obtenção de um grafo no interior de uma forma, detalhes podem ser vistos nos trabalhos de [64] e [65].

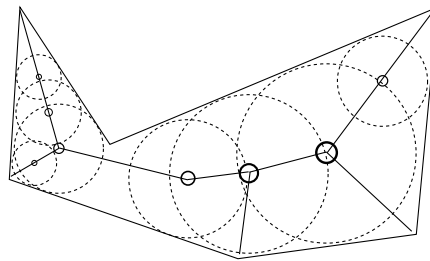


Figura 18: Obtenção do grafo interior a uma forma

A Figura 19 mostra um exemplo dos grafos que são comparados utilizando essa técnica. Mais detalhes podem ser encontrados em [66].

Outro tipo de abordagem é mostrada em [67] onde os autores utilizam a coleta de informações das bordas através de uma adaptação da técnica *Shape Context*, como mostrada em [15], através do que eles chamam de *Inner Distance Shape Context*. Essa técnica é regida pela aquisição de pontos que preenchem os *bins* observando o ângulo que tangencia o contorno nos pontos de articulação da imagem, pois é esse ângulo que dá o sentido de

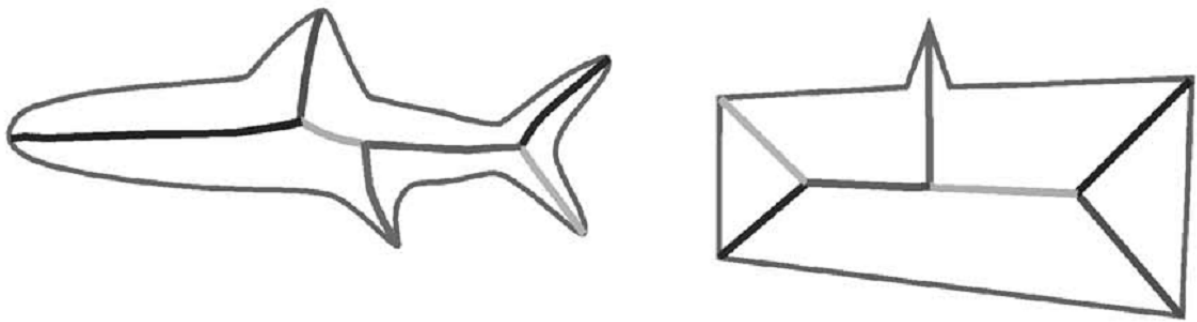


Figura 19: Duas formas e os grafos formados pelos seus eixos médios. Fonte: [4]

orientação para o gráfico log-polar. Essa abordagem é encontrada também no trabalho de [68].

No trabalho de [3], o autor dá uma contribuição sobre o reconhecimento de padrões utilizando estruturas auxiliares denominadas *Deformed Graphs*, os quais são usados para averiguar a mínima taxa de deformação sobre dois grafos em que se deseja efetuar uma associação.

Não obstante, em [3] é apresentada uma técnica fundamentada no paradigma estatístico do reconhecimento de padrões onde o autor usa um procedimento de mensagens entre os vértices executando um algoritmo com propagação de crenças - *belief propagation* - para avaliar a melhor associação. Para custear a associação entre vértices o autor utiliza a mesma função de custo usada na contribuição de vértices de sua implementação utilizando *Deformed Graphs*.

O algoritmo de propagação de crenças se baseia na troca de mensagens entre os vértices do grafo através do sistema de vizinhança constituído pelo conjunto de arestas. Essas mensagens são vetores que armazenam custos de associação entre um vértice de um grafo e os demais vértices de outro grafo. Mais detalhes sobre essa técnica podem ser encontrados em [69] e [70].

Os resultados dos trabalhos de [3] e [67] serão usados no capítulo 5 para comparação com os testes efetuados neste trabalho.

Nós procederemos com testes comparativos pelas técnicas de reconhecimento de formas desenvolvidas em nosso trabalho com os resultados disponíveis em [3] aplicados à bases de imagens COIL e MNIST e com os resultados disponíveis em [67] para a base de imagens KIMIA.

Os resultados dos testes efetuados bem como detalhes a respeito das instâncias das ba-

ses de imagens utilizadas são descritos no capítulo 5.

Levando em consideração o trabalho mostrado por [3] observamos que é possível tratar o problema de reconhecimento de formas através da associação de vértices em grafos considerando que na função de custo a informação da associação das arestas, induzida pela associação de vértices, é avaliada.

3.4 O problema de associação de pontos - PPM

Nosso trabalho trata o reconhecimento de formas em imagens através da associação de pontos e assim como muitos encontrados na literatura, a modelagem do problema é representada por associações entre grafos, os quais são obtidos por meio de aquisição de informações em imagens, por exemplo. Trabalhos que tratam do problema de associação de pontos podem ser encontrados em [71], [72], [73], [74] e [75].

Em [16] é descrito sucintamente o problema de casamento de padrões de pontos (PPM), do inglês *Point Pattern Matching*, o qual pode ser encarado como a tarefa de associar cada ponto de um conjunto a um determinado ponto de outro conjunto, tendo uma restrição ou um grupo de restrições a serem respeitadas e uma função objetivo, baseada em uma medida de similaridade, cujo resultado deve ser minimizado.

A Figura 20 mostra um exemplo de associação entre um conjunto e outro conjunto o qual é dado como base de informação. Nesse exemplo os padrões correspondentes são rotacionados em relação de um ao outro.

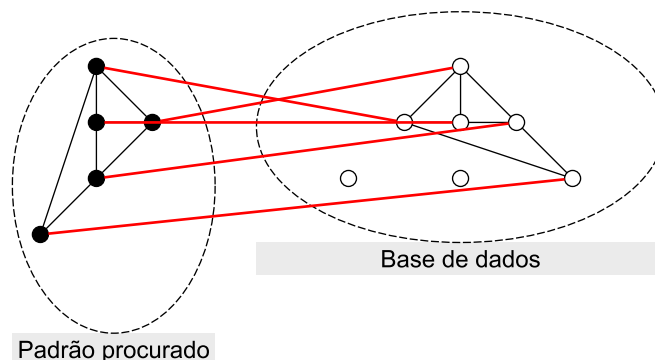


Figura 20: Exemplo de reconhecimento de padrões de pontos

O PPM pode ser exato, quando o conjunto de pontos “procurado” é idêntico a todo ou parte do conjunto de pontos da base de dados, ou inexato quando ocorre a presença de algum tipo de deformação no padrão existente na base de dados. Vale ressaltar que, o re-

conhecimento inexato é mais frequentemente encontrado em problemas da área de visão computacional e, para o caso do casamento inexato, os algoritmos propostos devem ser concebidos para operarem com algum grau de incerteza nas informações dos padrões e ainda apresentar robutez dada a possível presença de dados estruturais corrompidos nos padrões envolvidos ([16]).

Dadas as características do PPM como a necessidade de uma representação robusta e o armazenamento de informações estruturais, entre outras, o mesmo pode ser encarado como um problema de associação de grafos como descrito em [76] e apresentado em [77] e [78].

3.5 Modelagem do problema

Nossa abordagem, em suma, se configura como um problema quadrático de alocação, do inglês *Quadratic Assignment Problem* (QAP), o qual, segundo [79] foi introduzido em 1957 por Koopmans e Beckmann como sendo um modelo matemático para alocação de um conjunto indivisível de atividades econômicas, sendo sua formulação, como mostrada em [80], baseada na existência de duas matrizes $n \times n$ tais que, sejam C_{ij} uma matriz de custos, D_{ij} uma matriz de distâncias e, tendo um conjunto de inteiros $1, 2, \dots, n$ com S_n indicando o conjunto de permutações de $1, 2, \dots, n$ entre as matrizes, o QAP é definido como

$$\min \sum_{i=1}^n \sum_{j=1}^n c_{ij} d_{\pi(i)\pi(j)} \quad (3.1)$$

sobre todas as permutações $\pi \in S_n$.

Da interpretação dada em [81], onde os autores descrevem o QAP como um problema NP-Completo citando [82], o QAP pode ser entendido como um problema onde se possui um conjunto com m departamentos os quais se deseja alocar a n locações fazendo com que, para cada locação apenas um departamento seja alocado e, analogamente, cada departamento é alocado em apenas uma locação.

No trabalho de [80] o PPM é interpretado como a tarefa de alocar um conjunto de instalações num conjunto de localidades tendo o custo atribuído a esta tarefa como sendo uma função em relação às distâncias e fluxos de movimentação entre as instalações adicionado ao custo de se implantar determinada instalação em uma certa localidade.

Especificando assim, tem-se três matrizes $n \times n$, $Q = q_{ij}$, $D = d_{kl}$ e $B = b_{ik}$, que representam o fluxo de movimentação (q_{ij}) entre a instalação i e j , a distância (d_{kl}) entre a localidade k e a localidade l e o custo para a implantação (b_{ik}) da instalação i na localidade k . Dessa

forma, sendo n o número de instalações e localidades e dado o conjunto $N = \{1, 2, \dots, n\}$, temos que o QAP se refere a

$$\min_{\pi \in S_n} \sum_{i=1}^n \sum_{j=1}^n q_{ij} d_{\pi(i)\pi(j)} + \sum_{i=1}^n b_{i\pi(i)} \quad (3.2)$$

onde S_n é o conjunto de todas as permutações $\pi : N \rightarrow N$ e cada produto individual $q_{ij} d_{\pi(i)\pi(j)}$ representa o custo para associar uma instalação i a uma locação j [79].

Em seu trabalho [3] utiliza uma modelagem semelhante para formular uma de suas aplicações para o casamento de pontos, e é esta modelagem que iremos utilizar em nosso trabalho.

De fato, pode-se fazer uma analogia a este tipo de aplicação onde cada ponto de um conjunto (G_i nesse caso) é encarado como sendo o departamento e cada ponto em G_m seria visualizado como a localidade onde se deseja fazer a alocação.

Essa analogia pode ser exemplificada pela Figura 21 na qual são exibidos dois grafos que representam o conjunto de instalações e o conjunto de alocações, respectivamente em 21(a) e 21(b), que são visualmente encarados como dois grafos, G_i e G_m , cujos vértices devem ser associados. As arestas de G_i nesse caso são os fluxos de movimentação entre as instalações e as arestas de G_m são as distâncias entre cada alocação. O custo de implantação se dá diretamente pela associação entre um vértice de G_i e um de G_m e completando a analogia, temos na Figura 22 os custos de implantação para a instalação 2 em cada localidade representada por letras. Os demais custos foram omitidos para permitir uma melhor visualização da ideia que queríamos mostrar.

3.5.1 Modelagem para o problema de associação de pontos através da associação de grafos

Neste trabalho usaremos a concepção de grafos de atributos relacionais, do inglês *Attributed Relational Graph* (ARG) segundo a notação dada em [83] e caminharemos com um procedimento de reconhecimento estrutural de padrões.

Dado $\tilde{G} = (V, E)$ um grafo dirigido onde V representa o conjunto de vértices e E representa o conjunto de arestas, sendo que dois vértices $a, b \in V$ são ditos adjacentes se a aresta $(a, b) \in E$, um ARG é definido como $G = (V, E, \mu, \omega)$, onde $\mu : V \rightarrow L_V$ representa um vetor de atributos para cada vértice de V e, similarmente, $\omega : E \rightarrow L_E$ representa um vetor de atributos para cada aresta de E .

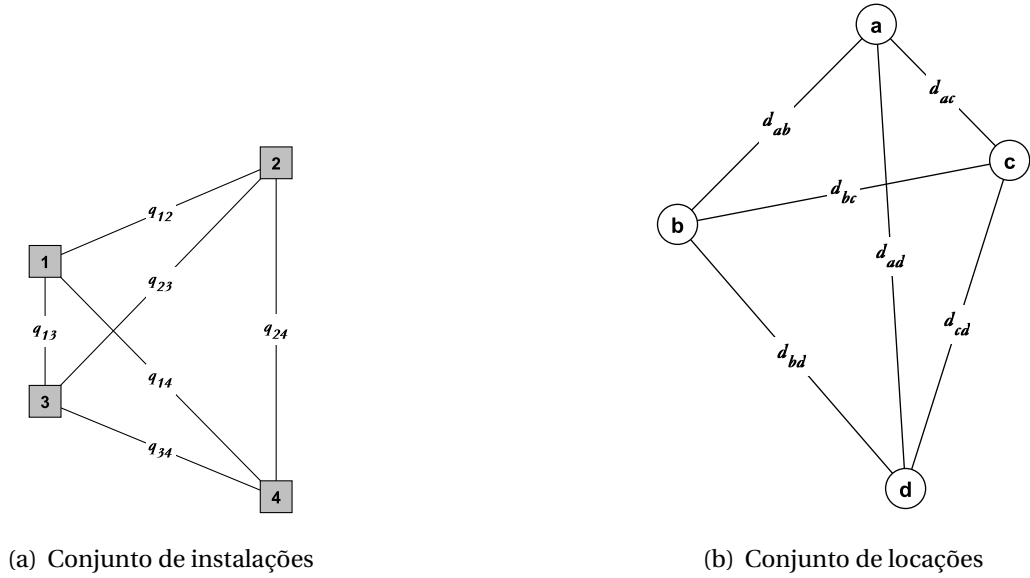


Figura 21: Analogia: **21(a)** é encarado como G_i e deve ser associado a **21(b)** que é encarado como G_m

Utilizamos aqui o conceito de reconhecimento baseado em modelos e, assim como em [3] e [83], faremos uma comparação entre dois ARG's, $G_i = (V_i, E_i, \mu_i, \omega_i)$ denotado como entrada, para representar a imagem que se deseja efetuar o reconhecimento, e chamamos de modelo o grafo $G_m = (V_m, E_m, \mu_m, \omega_m)$.

Para simplificarmos a referência às estruturas aqui utilizadas denotaremos, assim como em [3], ARG apenas por grafo, assim sendo, $|V_i|$ indica o número de vértices do grafo de entrada e $|E_i|$ indica o número de arestas. Analogamente, $|V_m|$ indica o número de vértices do grafo modelo e $|E_m|$ indica o número de arestas do mesmo. Dessa forma, o que buscamos na associação de grafos é encontrar o mapeamento $g: V_i \rightarrow V_m$ que represente a associação entre os vértices do grafo de entrada e do modelo.

A ideia central ao se usar uma modelagem do problema baseada em atribuição quadrática está relacionada ao fato de se buscar uma mensuração quanto a contribuição de vértices (termo linear) e a contribuição das arestas (termo quadrático).

Em [3] o termo linear representa a distância de aparência (d_A), cuja definição é condicionada ao problema abordado. Por exemplo, a contribuição de características de cores para medir a dissimilaridade entre um vértice e outro no grafo (o que pode ser encarado sobre as diferenças entre o tom de cinza que caracteriza um vértice e o tom de cinza de outro vértice), o que não é o caso do nosso trabalho. Em nosso trabalho usaremos informações sobre o posicionamento geométrico dos vértices, dada a sua disposição dentro da imagem, para avaliar a contribuição de aparência no custo de nossa função.

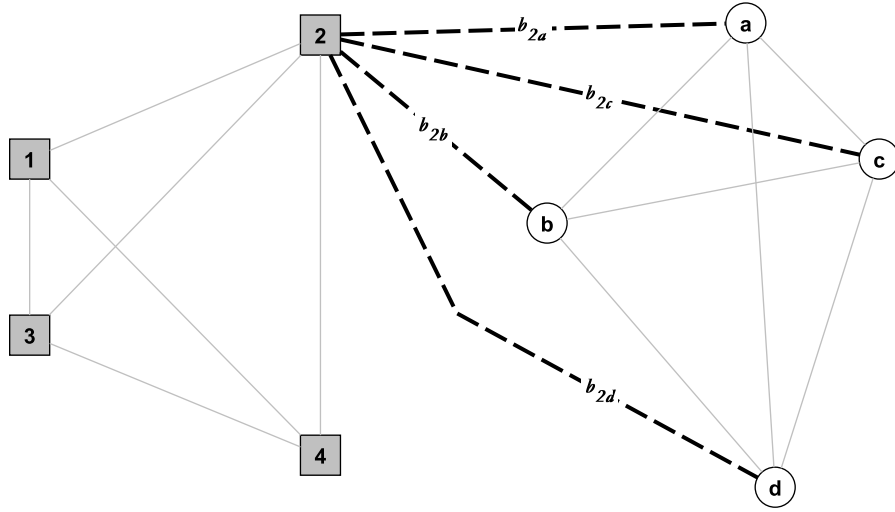


Figura 22: Analogia: custos de implantação da instalação nas localidades referentes

Além do termo linear, o termo quadrático é usado como uma função de avaliação estrutural do grafo a partir de suas adjacências. Esse termo é denotado por distância estrutural (d_S).

Dado o exposto, a associação entre grafos é mensurada pela minimização de uma função de custo proposta em [3] e reproduzida na equação 3.3:

$$C(g) = \lambda_1 \sum_{v_i \in V_i} d_A(v_i, g(v_i)) + (1 - \lambda_1) \sum_{(v_i, v'_i) \in E_i} d_S((v_i, v'_i), (g(v_i), g(v'_i))) \quad (3.3)$$

A equação 3.3 guarda similaridades em relação à equação 3.2 onde apontamos semelhanças de modo com que o termo $d_A(v_i, g(v_i))$ da equação 3.3 é equiparado ao termo $b_{i\pi(i)}$ da equação 3.2 por representarem custos diretos relacionados à sobreposição de um vértice por outro e os termos $d_S((v_i, v'_i), (g(v_i), g(v'_i)))$ e $q_{ij}d_{\pi(i)\pi(j)}$, de suas respectivas equações, são equiparados por tratarem custos relacionados às estruturas de vizinhança de cada vértice.

Do exposto, ao utilizarmos grafos como representação dos objetos, inferimos que o nosso objetivo é conseguir um mapeamento representado por um conjunto de associações entre o conjunto de vértices do grafo de entrada e o conjunto de vértices do grafo modelo (proveniente do banco de dados de imagens utilizado) que minimize a equação 3.3. Para procedermos com essa associação utilizamos não só informações provenientes dos vértices mas também das arestas do grafo, informações estas que são tratadas como informações estruturais e constituem o termo quadrático da equação 3.3.

A modelagem aqui proposta usa uma analogia entre o QAP e o que fora proposto em [33] onde a autora utiliza parcelas de contribuições entre os custos de associação de vértices e os custos de associação de arestas e aqui, da mesma forma, nosso objetivo é minimizar uma função de custo onde avaliamos a associação entre vértices e a associação entre arestas. Dessa forma vamos proceder nos referindo sobre o custo de associação de vértices apenas por contribuição de vértices, representado por c_v , e o custo de associação de arestas, representado por c_e , como a contribuição das arestas na Equação 3.4:

$$f = \alpha \sum c_v + (1 - \alpha) \sum c_e \quad (3.4)$$

Analogamente à função objetivo do QAP, o termo linear corresponde à contribuição dos vértices, enquanto o termo quadrático corresponde à contribuição das arestas, na função de avaliação f . No próximo capítulo serão apresentadas várias propostas para atribuir a função de avaliação (custos c_v e c_e) neste trabalho.

4 *Estratégias de Resolução*

Neste capítulo explicitaremos as técnicas empregadas neste trabalho para a tarefa de reconhecer formas em imagens através da associação de grafos tomando como base o casamento entre pontos onde, dados G_i e G_m , o resultado desejado é que seja retornado ao usuário um mapeamento um para um de todos os vértices do grafo de entrada para os vértices do grafo modelo.

Ainda neste capítulo serão detalhados os algoritmos propostos para serem aplicados em três bases de testes diferentes, procedendo assim com um trabalho de comparação de uma imagem de entrada, sem classificação, com imagens pertencentes a um banco de imagens para averiguar a que classe a imagem de entrada deve ser classificada.

Para as informações estruturais das arestas propomos um método que avalia dois critérios, um local, onde são avaliadas informações de comparação entre duas arestas como a diferença de tamanho e rotação entre elas, e um critério global que avalia a posição relativa da aresta dentro do grafo.

4.1 Representação da associação entre grafos

Em nosso trabalho a associação é representada por uma estrutura simples sendo uma matriz de inteiros com duas colunas e com $|V_i|$ linhas, onde cada valor na segunda coluna representa o identificador do vértice correspondente do grafo modelo e, ainda, com o vértice do grafo de entrada indicado pelo número presente na primeira coluna. Um exemplo desse mapeamento é exibido na Tabela 1 onde a primeira coluna mostra os identificadores dos vértices do grafo de entrada e a segunda coluna mostra os valores que representam os identificadores dos vértices do grafo modelo.

Nesse exemplo, temos a situação fictícia em que o vértice de índice 0 (v_{i0}) do grafo de entrada (G_i) está associado ao vértice de índice 4 (v_{m4}) do grafo modelo (G_m), v_{i1} com v_{m3} e assim por diante conforme a sequência apresentada. A Figura 23 mostra a visualização

Tabela 1: Exemplo de matriz de associação

0	4
1	3
2	0
3	1
4	5
5	2

gráfica desse exemplo onde os círculos brancos representam os vértices do grafo de entrada e os círculos cinza são os vértices do modelo. Essa representação será útil na seção 4.4.2 onde apresentamos um método de melhoria de solução aplicado sobre as associações entre os grafos.

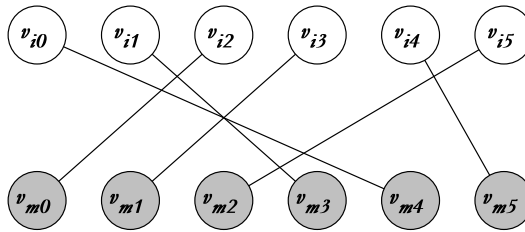


Figura 23: Exemplo de associações geradas

4.2 Custo da associação entre grafos

Neste trabalho, o problema de interesse trata do mapeamento de G_i em G_m visando associar todos os vértices do grafo de entrada com vértices do grafo modelo. Quando isso não for possível (caso em que $|V_i| > |V_m|$ por exemplo) os vértices não associados receberão um custo de penalização simbolizando que os mesmos encontram-se desassociados. Desta forma, a representação das associações será como o exposto na Figura 24 onde o grafo modelo possui um vértice a menos que o grafo de entrada e, portanto, temos um caso em que um vértice de V_i (no exemplo o vértice v_{i4}) está desassociado, o qual recebe custo de associação igual a c_p que representa a penalidade atribuída. Essa penalidade é o pior valor obtido entre as associações válidas ao final do processo algorítmico.

O termo $c(v_i, v_m)$ indica o custo geral de associação entre os vértices $v_i \in V_i$ e $v_m \in V_m$

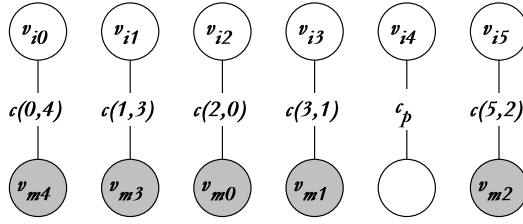


Figura 24: Exemplo de associações com penalização

dado pela equação 4.1.

$$c(v_i, v_m) = \alpha c_v(v_i, v_m) + (1 - \alpha) c_e(v_i, v_m) \quad (4.1)$$

Das Equações 4.1 e 3.4 temos que o objetivo do nosso algoritmo é minimizar o somatório da Eq. 4.2 onde o termo c_e que avalia a contribuição das arestas é calculado pela comparação entre as arestas adjacentes aos vértices v_i e v_m em seus respectivos grafos.

$$\sum c(v_i, v_m) = \sum [\alpha c_v(v_i, v_m) + (1 - \alpha) c_e(v_i, v_m)] \quad (4.2)$$

A equação 4.1 é usada para atribuir os custos das associações entre vértices e é comum a todos os algoritmos aqui implementados assim, nosso objetivo é estabelecer as funções de avaliação c_v e c_e a serem aplicadas para cada base de imagens.

4.3 Funções de avaliação

Os algoritmos implementados neste trabalho são estratégias classificadoras baseadas em comparações com protótipos que são imagens contidas em bancos de imagens como explicado na seção 2.3.2.3.

A função para custear a associação entre um vértice de G_i e um vértice de G_m trabalha com dois tipos de avaliação: a contribuição de vértices c_v (termo linear) e a contribuição de arestas c_e (termo quadrático), para custear uma associação entre dois grafos.

As funções de avaliação aplicadas a cada termo c_v e c_e são definidas nessa seção.

4.3.1 Contribuição de vértices - Termo linear

Para custear a contribuição de vértices desenvolvemos dois métodos baseados em *Shape Context* que serão descritos nas duas próximas subseções.

4.3.1.1 *Shape Context Percentual - SCP*

Com a finalidade de avaliar o custo da contribuição de vértices propomos uma adaptação do que fora implementado por [3] para o custo da distância de aparência (que foi a nomenclatura usada para a contribuição de vértices) de suas aplicações com relação ao *Shape Context* apresentado em [15], [84] e [85].

A partir de uma imagem, a técnica original diz respeito a um histograma que é a representação gráfica da quantidade de pontos em *bins* obtidos através da sobreposição do gráfico log-polar (onde se tem divisões quanto ao raio e ângulos) sobre cada ponto. A Figura 25 exemplifica um disco log-polar com três subdivisões de raio e oito angulares somando assim 24 *bins* os quais aparecem enumerados de 01 a 24.

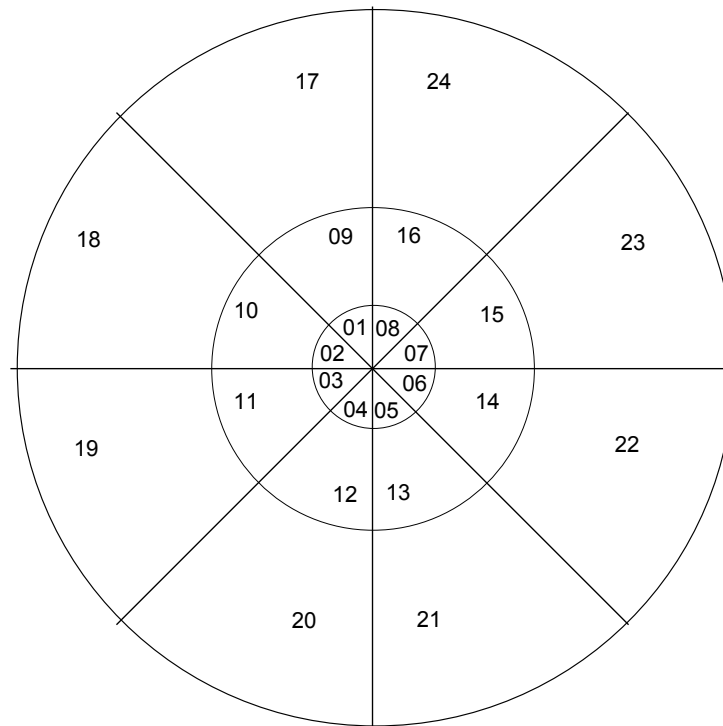


Figura 25: Gráfico log-polar

Com isso, cada *bin*, que é cada divisão desse gráfico, gera um valor numérico indicando a contagem de pontos presentes dentro desse *bin*. Esse valor é armazenado em um vetor para, ao final, computar um histograma. Dessa forma, quando comparados dois grafos, dado um

vértice v_i do primeiro grafo (de entrada), é calculado o histograma h_i relativo às coordenadas dos demais vértices presentes no grafo, conforme a Equação 4.3.

$$h_i = \#\{v \neq v_i : (v - v_i) \in \text{bin}(k)\} \quad (4.3)$$

Este histograma é definido como o *shape context* de v_i . Considerando o vértice v_i do grafo de entrada e o vértice v_m do grafo modelo o custo de associação C_{ij} é dado pelo teste estatístico χ^2 (qui-quadrado) conforme a equação 4.4:

$$C_{im} \equiv C(v_i, v_m) = \frac{1}{2} \sum_{k=1}^K \frac{[h_i(k) - h_m(k)]^2}{h_i(k) + h_m(k)} \quad (4.4)$$

Nessa equação $h_i(k)$ e $h_m(k)$ denotam o k -ésimo histograma normalizado do vértice v_i e v_m , respectivamente.

Um exemplo simples do cálculo efetuado pode ser visto na Figura 26 onde são ilustradas duas aquisições de pontos, a primeira em G_i e a segunda em G_m . O cálculo para este caso, onde o único *bin* preenchido em G_i possui 10 pontos no seu interior e seu correspondente em G_m não possui nenhum ponto em seu interior, é tal que, seu custo vale $\frac{(10-0)^2}{10} = 10$, e da fórmula 4.4 temos que o valor do C_{ij} para esse caso é $\frac{10}{2} = 5$.

Os histogramas referentes às tomadas de pontos ilustradas na Figura 26 são mostrados na Figura 27(b) sendo que o histograma ilustrado em 27(a) é referente à tomada de pontos mostrada em 26(a) e o ilustrado em 27(b) é referente à tomada de pontos mostrada em 26(b).

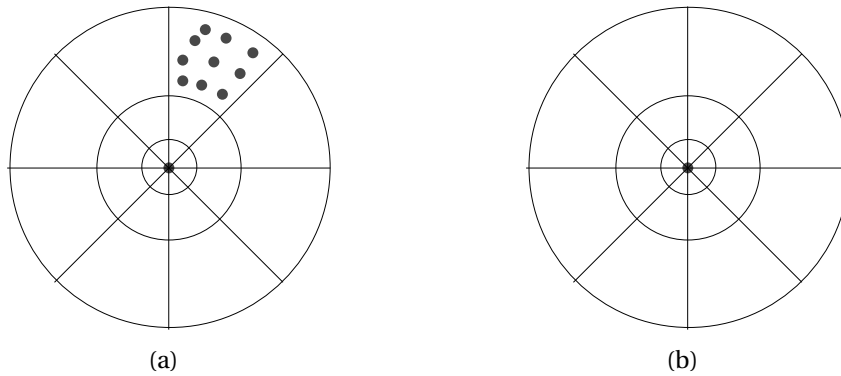


Figura 26: Exemplo para cálculo de C_{ij} : a imagem em 26(a) representa tomada de pontos para o histograma de um ponto de G_i e a imagem em 26(b) se refere à tomada de pontos em G_m

A nossa abordagem difere da formulação original, onde não usamos a medida estatística χ^2 mas sim a diferença percentual entre os valores numéricos armazenados relativos a cada

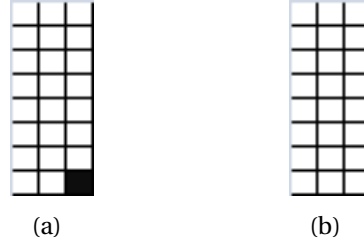


Figura 27: A imagem em 26(a) gera o histograma 27(a) e a imagem em 26(b) gera o histograma 27(b)

bin, obtendo, ao final uma média dessas diferenças. O objetivo único nesse caso é a obtenção um resultado numérico final pertencente ao conjunto de valores $[0 \dots 1]$. Dessa forma o termo linear C'_{im} é calculado pela equação 4.5.

$$C'_{im} \equiv C'(v_i, v_m) = \frac{1}{K} \sum_{k=1}^K \frac{|h_i(k) - h_m(k)|}{\max(h_i(k), h_m(k))} \quad (4.5)$$

Com isso, o exemplo dado na Figura 26 teria seu cálculo com o custo de $\frac{|10-0|}{10} = 1$, ou 100% de diferença. Daí, o valor de C'_{ij} para esse caso será $\frac{1}{24}$, dado que a soma dos outros *bins* é zero.

Nos referiremos à essa adaptação como *SCP*, abreviatura de *Shape Context Percentual*. Ressaltamos ainda que o exemplo dado nessa seção é fictício e tem caráter apenas informativo, nas nossas implementações utilizamos as mesmas configurações de [15] tendo 5 divisões de raio e 12 divisões angulares totalizando 60 *bins*.

Assim, o cálculo do *SCP* para comparação entre um vértice v_i do grafo de entrada e um vértice v_m do grafo modelo é dado pela Equação 4.6:

$$SCP(v_i, v_m) = C'(v_i, v_m) = \frac{1}{K} \sum_{k=1}^K \frac{|h_i(k) - h_m(k)|}{\max(h_i(k), h_m(k))} \quad (4.6)$$

4.3.1.2 Shape Context para arestas - SCA

Um outro tipo de avaliação para a contribuição de vértices leva em consideração a existência de aresta(s) conectada(s) ao vértice em que se deseja adquirir informações.

A idéia inicial é proveniente do que foi proposto em [29] baseado em [5], ou seja, uma adequação da técnica de *Hashing Geométrico* (HG), a qual trabalha com uma fase *offline* para proceder com as transformações necessárias para obtenção das bases que formam a tabela *hash* daquela técnica e uma fase *online* onde se estabelece o reconhecimento de uma

base dada como entrada.

Na fase *offline* do HG, cada aresta se transforma em uma base de comprimento unitário através de transformações geométricas (rotação, translação e escalonamento) no plano 2D e as informações a respeito dos demais pontos são armazenadas constituindo um banco de informações de histogramas que é utilizado na fase *online*, ou seja, na fase de comparação. Mais detalhes a respeito do *Hashing Geométrico* podem ser obtidos em [86].

Na Figura 28 é ilustrado um conjunto de pontos dado como modelo, o qual é submetido a essas transformações geométricas e posicionado sobre o eixo horizontal para a aquisição dos pontos para a composição dos respectivos histogramas. No exemplo mostrado na figura, temos a base para construção da tabela hash formada pelos pontos v_6 e v_4 .

De acordo com a ilustração da Figura 28, primeiramente é passado o conjunto de pontos original que, em seguida, sofre transformações geométricas de escalabilidade a fim de que a distância entre os pontos que se deseja captar informações (no caso v_6 e v_4) seja unitária. Após esse processo de escalonamento, uma operação de rotação e translação é efetuada sobre todos os pontos do conjunto a fim de que o vértice v_6 esteja sobre o ponto de coordenadas $(-\frac{1}{2}, 0)$ e v_4 esteja sobre o ponto de coordenadas $(\frac{1}{2}, 0)$, só nesse momento as informações a respeito da tabela são coletadas.

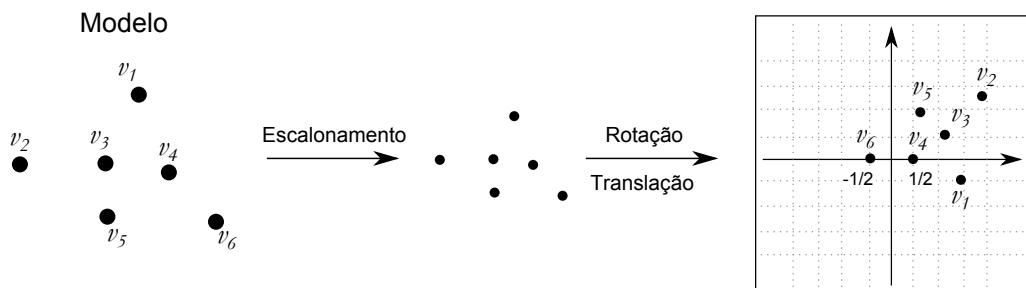


Figura 28: Tomada de base - HG. Fonte:[5], com adaptação

A Figura 30 dá um exemplo do que foi proposto inicialmente como diferencial da técnica original. Cada aresta ilustrada na Figura 29, sofre as transformações geométricas sugeridas pelo *Hashing Geométrico*, contudo, não há a fase de pré-processamento *offline* e a avaliação é feita com base na comparação direta das arestas vizinhas àquelas selecionadas.

Após o processo de escalonamento, rotação e translação das arestas de entrada e do modelo sobre um eixo imaginário, os vértices das arestas vizinhas são comparados a partir da distância euclidiana dos vértices extremos das arestas vizinhas à aresta escolhida para formar a base.

Observando essa proposta e atentando para o que foi relatado em [3] quanto às infor-

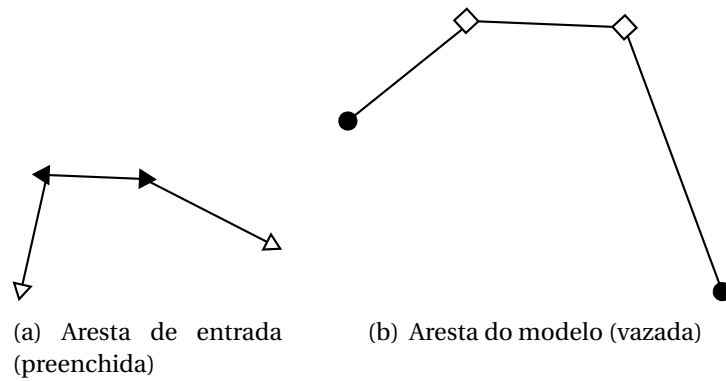


Figura 29: Arestas de entrada e do modelo com suas respectivas vizinhas

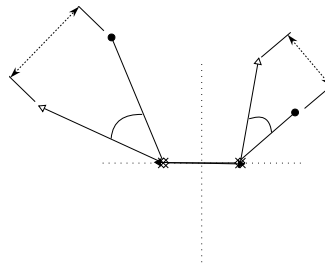


Figura 30: Comparação de arestas - HG

mações de aparência das imagens, notamos que a técnica, empregada dessa forma, não resultaria em bons frutos dado que, como a comparação proposta não trabalhava com o pré-processamento original, informações preciosas a respeito do posicionamento real dos vértices no grafo seriam perdidas nos processos de transformações geométricas além de não serem comparados todos os pontos do grafo mas apenas os vizinhos à aresta selecionada.

Neste trabalho queremos evitar a fase *offline*. Desta forma, procederemos com as transformações não no conjunto de pontos mas sim na transformação geométrica do gráfico log-polar antes da obtenção dos pontos que irão constituir nossos histogramas. A partir desse estudo, adaptamos o método levando em consideração o posicionamento das arestas em relação aos seus vértices extremos.

A tomada dos pontos no que se refere ao *Shape Context* para Arestas - *SCA* - é dada pelo ponto médio de cada aresta, onde é feita uma rotação no disco log-polar no momento da aquisição dos dados. Essa rotação do disco é feita levando-se em consideração o sentido da aresta em questão, ou seja, dada uma aresta (a, b) qualquer, os eixos do gráfico log-polar devem estar perpendiculares a esta aresta conforme mostrado na Figura 31.

Por exemplo, tomando como base a aresta (6, 7) do grafo de exemplo mostrado na Figura 32, a tomada de pontos faz com que o eixo das ordenadas tenha sentido de crescimento perpendicular à aresta, assim como é mostrado na Figura 33(c).

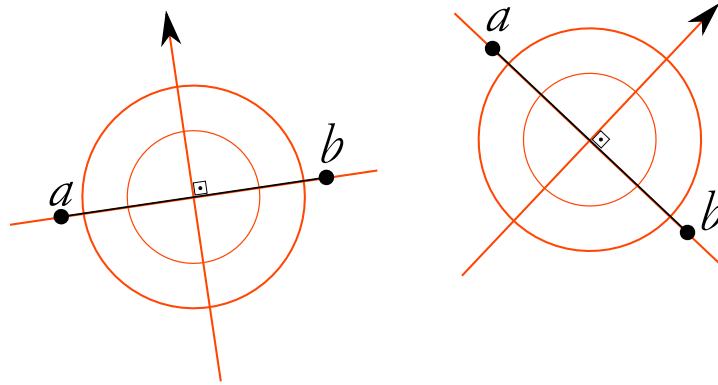


Figura 31: Perpendicularidade dos eixos do grafo log-polar

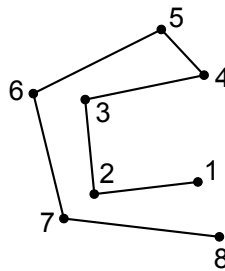


Figura 32: Exemplo de grafo

A Figura 33(a) mostra a tomada de dados da técnica SCP. Na sequência, nós temos a centralização dos eixos do grafo log-polar coincidindo com o ponto médio da aresta escolhida como exemplo (Figura 33(b)) e o efeito de rotação mostrado na Figura 33(c).

Dessa forma, inferimos que o cálculo do SCA pode ser expresso como uma função que recebe como parâmetros duas arestas e_i e e_m , de entrada e do modelo, respectivamente, conforme a Equação 4.7. O pseudocódigo de cálculo do SCA está disposto no Algoritmo 1.

$$SCA(e_i, e_m) = SCP(pMedio(e_i), pMedio(e_m)) \quad (4.7)$$

Algoritmo 1: Cálculo do custo SCA entre e_i e e_m

Dados: e_i, e_m

Resultado: c_{SCA}

- 1 $pm_i \leftarrow pontoMedio(e_i);$
 - 2 $pm_m \leftarrow pontoMedio(e_m);$
 - 3 $c_{SCA} \leftarrow SCP(pm_i, pm_m);$
-

O pseudocódigo descrito no Algoritmo 1 mostra como o SCA é calculado para duas arestas e segue os seguintes procedimentos em suas linhas:

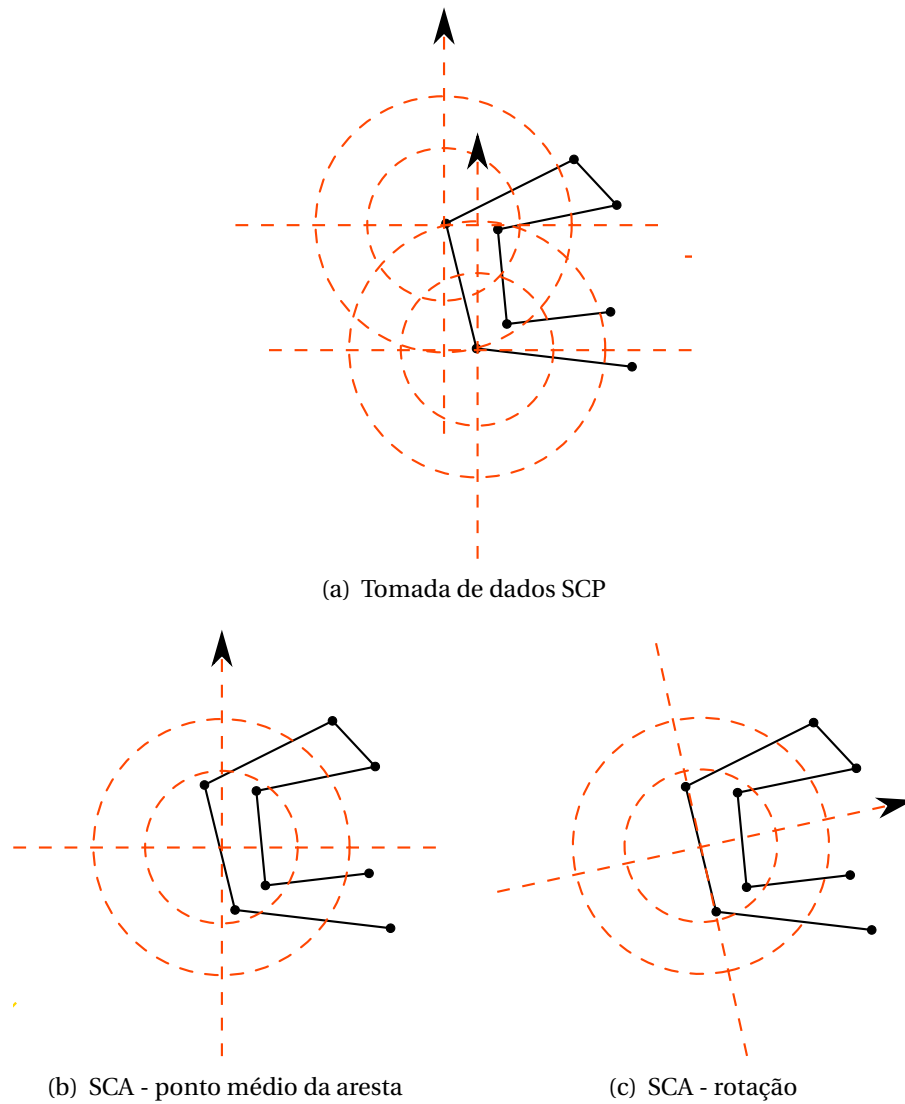


Figura 33: Diferenças da técnica original e a adaptada

- Linha 1: são calculadas as coordenadas do ponto médio referente à aresta e_i .
- Linha 2: são calculadas as coordenadas do ponto médio referente à aresta e_m .
- Linha 3: é calculado o custo SCP , como na Eq. 4.6, para os pontos médios de cada aresta e este é atribuído como sendo o custo SCA da função.

Nesta abordagem, procederemos com a aquisição das informações referentes ao SCA através das arestas ligadas ao vértice que será usado como comparação.

Visualmente, nós temos que a técnica procede como no exemplo mostrado nas Figuras 34 e 35 para a tomada de informações sobre a aresta (v_{i0}, v_{i4}) e, como neste exemplo o ponto em questão para a tomada de dados é o vértice v_{i0} . A próxima aquisição de informações é em respeito à aresta (v_{i0}, v_{i1}) já que é a outra aresta que tem o vértice v_{i0} como extremo.

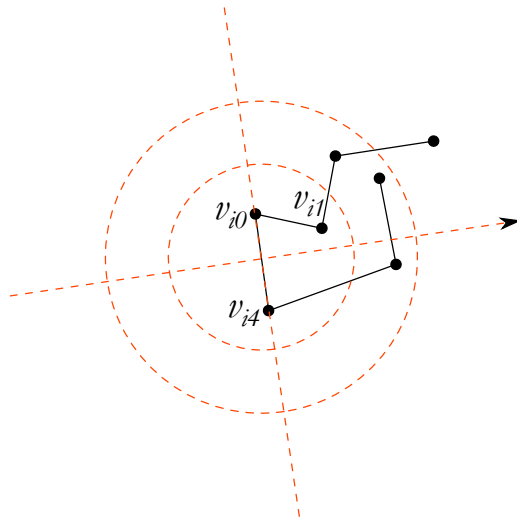


Figura 34: Tomada de dados SCA, aresta 01

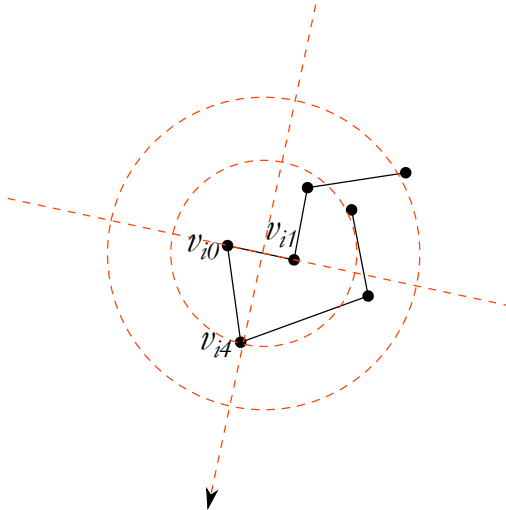


Figura 35: Tomada de dados SCA, aresta 02

A aquisição de informações para a criação dos histogramas em relação aos *bins* procede da mesma forma que no *SCP* (definido na seção 4.3.1.1) só que agora com o eixo do gráfico log-polar rotacionado.

Caso um ponto v_i tenha aresta incidente e outro v_m , o qual está se verificando a possibilidade de associação, não possua, o SCA não é calculado. A situação é análoga, quando um vértice v_m tem aresta incidente e v_i não.

O custo SCA_V atribuído na comparação entre dois vértices é avaliado como a média aritmética dos resultados do SCA da Eq. 4.7 para as duas arestas. Dessa forma teremos que, dadas duas arestas e_{i1} e e_{i2} ligadas a um vértice $v_i \in V_i$ e outras duas arestas e_{m1} e e_{m2} ligadas a um vértice $v_m \in V_m$ o cálculo do SCA_V procede conforme a Eq. 4.8:

$$SCA_V(v_i, v_m) = \begin{cases} \frac{SCA(e_{i1}, e_{m1}) + SCA(e_{i2}, e_{m2})}{2} & , \quad SCA(e_{i1}, e_{m1}) \neq 0 \wedge SCA(e_{i2}, e_{m2}) \neq 0 \\ SCA(e_{i1}, e_{m1}) & , \quad SCA(e_{i1}, e_{m1}) \neq 0 \wedge SCA(e_{i2}, e_{m2}) = 0 \\ SCA(e_{i2}, e_{m2}) & , \quad SCA(e_{i1}, e_{m1}) = 0 \wedge SCA(e_{i2}, e_{m2}) \neq 0 \\ 0 & , \quad SCA(e_{i1}, e_{m1}) = 0 \wedge SCA(e_{i2}, e_{m2}) = 0 \end{cases} \quad (4.8)$$

Como podemos observar na Equação 4.8, caso tenha apenas um termo calculado (apenas um par de arestas comparado) o SCA_V é dado pelo valor desse termo apenas.

Vale ressaltar que, com o emprego dessa técnica, o que esperamos é amenizar os efeitos da rotação nas comparações que são encontradas entre imagens. Um exemplo do que esperamos é apresentado na Figura 36(a) onde temos a letra “A” representada como um grafo e a tomada do SCP para determinado ponto. Na Figura 36(b) temos o mesmo grafo agora sob efeito de rotação. Podemos observar que o SCP para o mesmo ponto não produz o mesmo resultado.

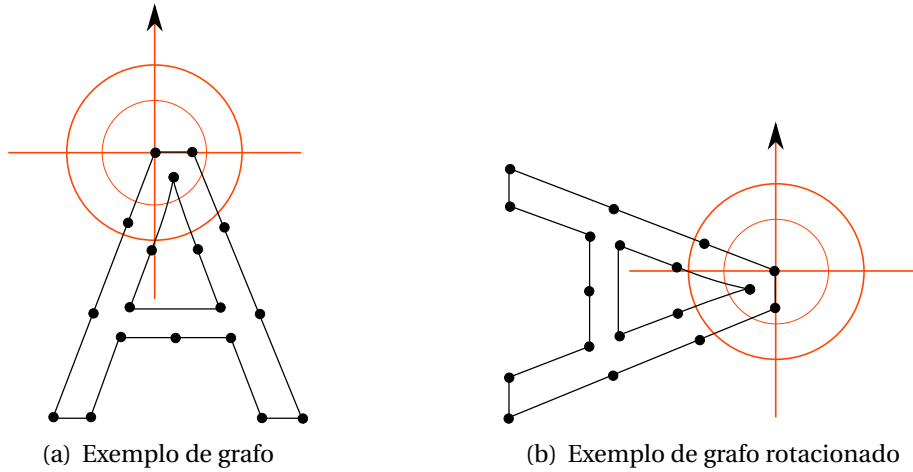


Figura 36: Exemplo de tomada de SC

Já no exemplo mostrado na Figura 37 mostramos o que estamos esperando de resultado, já que, com base na posição relativa da aresta em que se toma os dados (Figura 37(a)), após uma rotação, o sentido de aquisição dos dados para o histograma continua o mesmo, em relação ao grafo como um todo, como podemos observar na Figura 37(b).

Finalmente, a Equação 4.9 descreve o cálculo de contribuição de vértices desta abordagem. Ela é definida como um balanceamento, através do parâmetro λ (lambda), entre as funções descritas em 4.6 e 4.8.

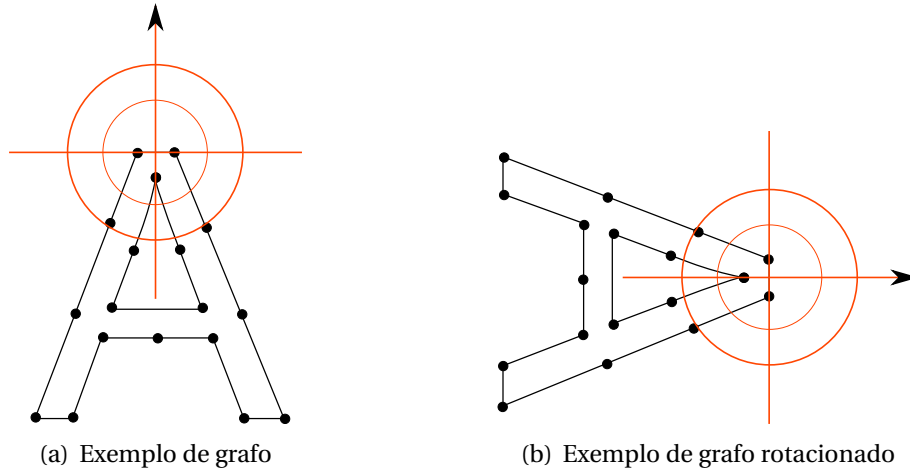


Figura 37: Exemplo de tomada de SC - adaptado

$$c_v(v_i, v_m) = \lambda SCP(v_i, v_m) + (1 - \lambda) SCA_V(v_i, v_m) \quad (4.9)$$

onde temos $0 \leq \lambda \leq 1$.

4.3.2 Comparação de arestas usando busca em profundidade

Os grafos que usamos em nosso trabalho são definidos como florestas, ou seja, formados por árvores que são componentes conexas sem ciclos [62].

Definimos ainda, devido ao tipo de imagem de uma das aplicações de interesse, que os graus dos vértices podem ser somente 0, 1 ou 2.

Valendo-nos dessa estrutura, avaliamos a contribuição das arestas tomando uma busca em profundidade aplicada aos grafos comparados que chamamos aqui de comparação em níveis.

Partindo de dois vértices dados como parâmetros de comparação, um de V_i e outro de V_m , essa busca em profundidade é efetuada conforme [87] através da formação do caminho do vértice escolhido, chamado de raiz, até as folhas da árvore ou até que um número de níveis de profundidade previamente indicado seja atingido.

Para facilitar o entendimento neste trabalho chamaremos a profundidade como nível e a avaliação se dá a cada comparação de vértices, onde é formado o caminho dado pela estrutura de arestas adjacentes que se inicia no vértice correspondente (raiz da subárvore). Essa subárvore é formada tanto a partir de v_i quanto a partir de v_m diretamente em seus respectivos grafos G_i e G_m e a comparação é feita a partir das arestas diretamente ligadas à

raiz (v_i e v_m) em direção às folhas das subárvores comparando duas arestas (uma de G_i e uma de G_m) que estejam na mesma profundidade na respectiva árvore. Esse procedimento é ilustrado na Figura 38, com a seleção dos vértices que serão comparados, na Figura 39, com a formação das subárvores em G_i e G_m , e na Figura 40 que representa a comparação entre arestas que estejam na mesma profundidade.

Através das subárvores formadas a partir de v_i (em G_i) e v_m (em G_m), a avaliação é feita comparando-se arestas duas a duas, uma de G_i e uma de G_m , que estejam no mesmo nível. Essa comparação é efetuada de acordo com a função de custo multicritério para arestas, c'_e , exibida na seção 4.3.3.

Como a comparação é feita em cada nível o processo de execução caminha para os adjacentes nas subárvores e as respectivas arestas de E_i e E_m são comparadas.

O número de níveis, N , para comparação é um parâmetro de entrada do algoritmo responsável pela execução desse procedimento. Caso a subárvore não alcance o número total de níveis estipulado para a comparação, caminha-se o máximo possível até que se encontre um nó folha (de grau um). No caso de existir ainda arestas, como é o caso da aresta e_{m5} ilustrada na Figura 40, a mesma é descartada para a comparação e não se atribui custo algum.

Esse procedimento é o mostrado no pseudocódigo presente no Algoritmo 2.

Os detalhes a respeito da execução do Algoritmo 2 são mostrados a seguir:

- Linha 1: inicializado o contador de avaliações efetuadas.
- Linha 2: inicializado o somador dos custos de avaliação de arestas.
- Linha 3: laço para a contagem de níveis à direita.
- Linha 4: seleciona uma aresta à direita em G_i que esteja ligada a v_i pela sua subárvore de adjacência.
- Linha 5: seleciona uma aresta à direita em G_m que esteja ligada a v_m pela sua subárvore de adjacência.
- Linha 6: verifica se ambas as arestas selecionadas não são nulas.
- Linha 7: se as arestas não são nulas então calcula-se c'_e e esse valor é somado ao valor de c_e
- Linha 8: é incrementado o contador de avaliações efetuadas.

Algoritmo 2: Cálculo do custo de associação de arestas em níveis

Dados: $v_i \in V_i, v_m \in V_m, \text{niveis}$
Resultado: c_e

```

1   $N = 0;$ 
2   $c_e = 0;$ 
3  para  $x = 1$  até  $\text{niveis}$  faça
4       $e_i \leftarrow \text{proximaArestaADireita}(v_i, x);$ 
5       $e_m \leftarrow \text{proximaArestaADireita}(v_m, x);$ 
6      se  $(e_i \neq \text{nulo})$  e  $(e_m \neq \text{nulo})$  então
7           $c_e = c_e + c'_e(e_i, e_m);$ 
8           $N = N + 1;$ 
9      fim
10 fim
11 para  $x = 1$  até  $\text{niveis}$  faça
12      $e_i \leftarrow \text{proximaArestaAEsquerda}(v_i, x);$ 
13      $e_m \leftarrow \text{proximaArestaAEsquerda}(v_m, x);$ 
14     se  $(e_i \neq \text{nulo})$  e  $(e_m \neq \text{nulo})$  então
15          $c_e = c_e + c'_e(e_i, e_m);$ 
16          $N = N + 1;$ 
17     fim
18 fim
19 se  $N > 0$  então
20     retorna  $\frac{c_e}{N};$ 
21 senão
22     retorna  $0;$ 
23 fim
  
```

- Linha 11: laço para a contagem de níveis à esquerda.
- Linha 12: seleciona uma aresta à esquerda em G_i que esteja ligada a v_i pela sua subárvore de adjacência.
- Linha 13: seleciona uma aresta à esquerda em G_m que esteja ligada a v_m pela sua subárvore de adjacência.
- Linha 14: verifica se ambas as arestas selecionadas não são nulas.
- Linha 15: se as arestas não são nulas então calcula-se c'_e e esse valor é somado ao valor de c_e .
- Linha 16: é incrementado o contador de avaliações efetuadas.
- Linha 19: verifica se o contador de avaliações é maior que zero.
- Linha 20: se o contador de avaliações for maior que zero então retorna o valor da média

$$\frac{c_e}{N}.$$

- Linha 22: se o contador de avaliações for igual a zero então retorna o valor igual a zero.

Tomando como exemplo os grafos ilustrados na Figura 38 onde são selecionados os vértices v_{i0} e v_{m0} para serem comparados, temos que os caminhos formados são os exibidos na Figura 39.

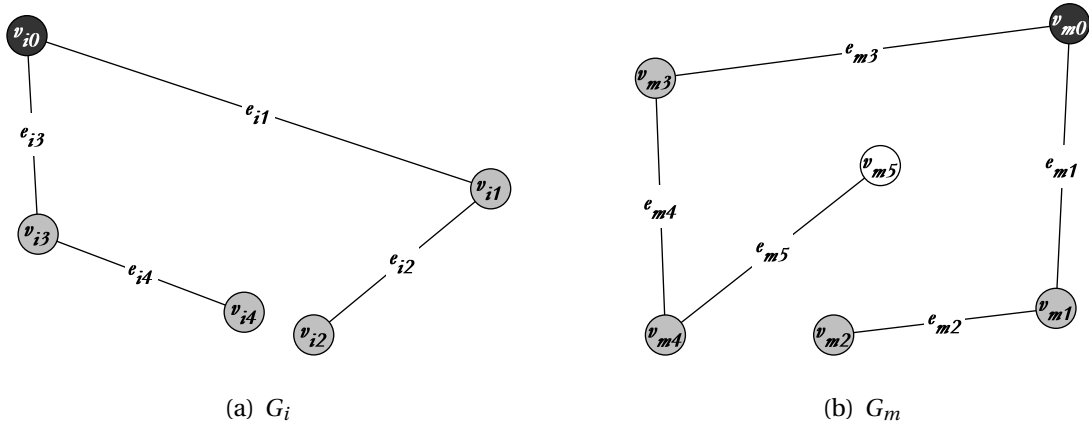


Figura 38: Vértices selecionados para comparação v_{i0} e v_{m0}

Na Figura 39(a) é ilustrado um exemplo de subárvore formada a partir do vértice $v_{i0} \in V_i$ e na Figura 39(b) temos um exemplo de subárvore formada a partir do vértice $v_{m0} \in V_m$. Por fim, a Figura 40 mostra um esquema de comparação em níveis, nesse caso temos dois níveis, para cada caminhamento a esquerda e a direita do vértice principal (raiz da subárvore).

Ressaltamos que esse tipo de abordagem foi possível dada a natureza das imagens aqui tratadas (relativas a uma das bases de imagens usada nos experimentos computacionais), as quais possuem características de contorno muito bem definidas, pois não haveria como usar esse tipo de tratamento para vértices com grau maior que 2.

4.3.3 Avaliação multicritério para arestas

Com a finalidade de averiguar a melhor comparação entre arestas duas a duas, foi proposta uma comparação onde são avaliados dois critérios, um atribuído às características locais e outro onde são avaliadas características globais em relação ao grafo a que as arestas pertencem.

A nomenclatura utilizada, local e global, se dá pelo fato de termos duas comparações distintas em relação à disposição das arestas, a saber:

- O critério local leva em consideração a comparação direta entre as arestas, avaliando,

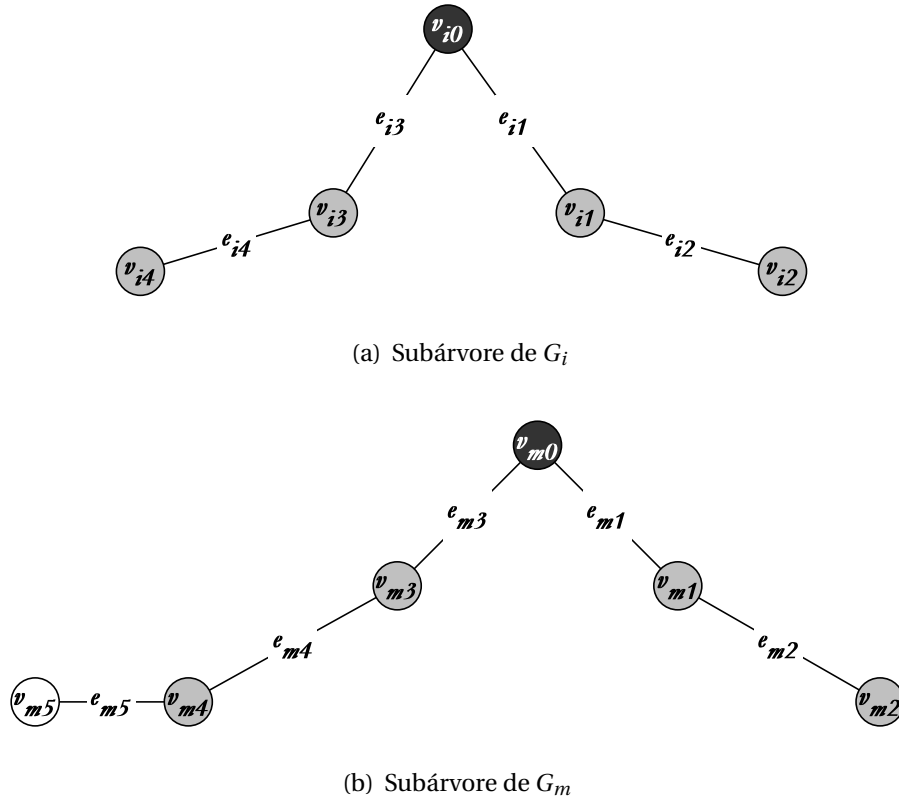


Figura 39: Arestas a serem comparadas - Vértices v_{i0} e v_{m0}

por exemplo, dados como comprimento e o ângulo formado dada a sobreposição de duas arestas em relação ao ponto médio de cada uma.

- O critério global avalia a posição relativa de cada aresta em relação ao grafo onde a mesma está inserida.

4.3.3.1 Critério local para avaliação de arestas - c_{eLocal}

A avaliação local considera uma comparação sobre as características geométricas de cada aresta, vistas como uma sobreposição em relação aos seus vértices iniciais.

Dessa forma, ao tomarmos duas arestas $e_i = (v_{i1}, v_{i2}) \in E_i$ e $e_m = (v_{m1}, v_{m2}) \in E_m$, o critério de avaliação local é dado avaliando a diferença percentual entre os ângulos das arestas em relação à posição vertical e, também, avaliando a diferença percentual entre o comprimento das duas arestas.

Para facilitar o entendimento preparamos o seguinte exemplo: tomemos as arestas e_i e e_m da Figura 41. Na Figura 42 são exibidos os ângulos β_i e β_m referentes a cada uma das arestas, e_i e e_m respectivamente, em relação ao eixo vertical.

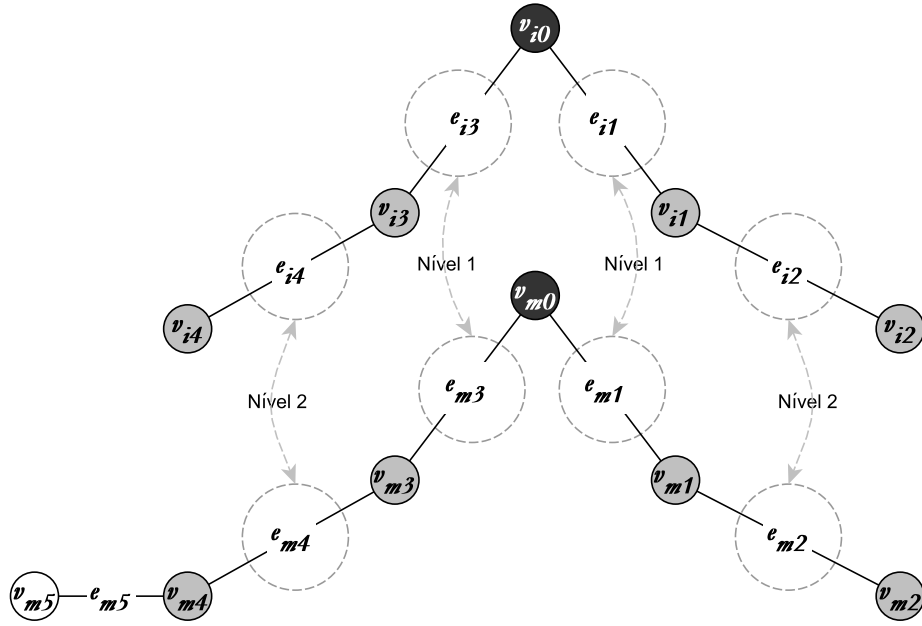


Figura 40: Comparação de arestas em níveis

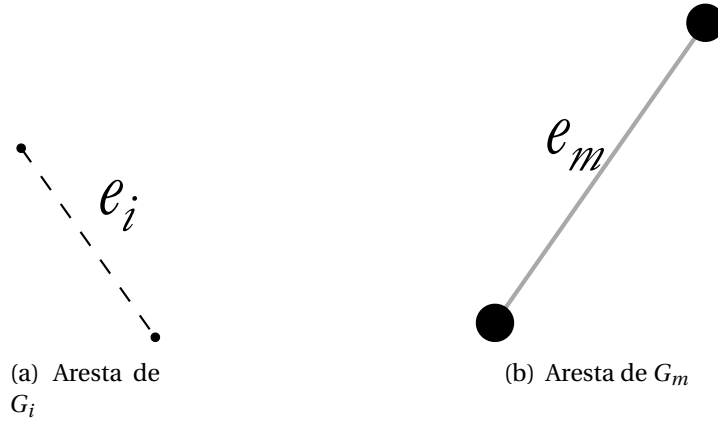


Figura 41: Arestas a serem comparadas - c_{eLocal}

Na Figura 43 representamos o comprimento de cada aresta, em unidades de medida, sendo d_i o comprimento referente a e_i e d_m referente a e_m .

Com essas informações, nosso critério de avaliação local é calculado tomando dois fatores, um angular e outro linear. O fator angular local é calculado segundo a Eq. 4.10:

$$f_{eAngularLocal}(e_i, e_m) = \begin{cases} \frac{|\beta_i - \beta_m|}{\text{Max}(\beta_i, \beta_m)} & , (\beta_i + \beta_m) \neq 0 \\ 0 & , (\beta_i + \beta_m) = 0 \end{cases} \quad (4.10)$$

O fator linear local é dado, de maneira análoga, pela Equação 4.11:

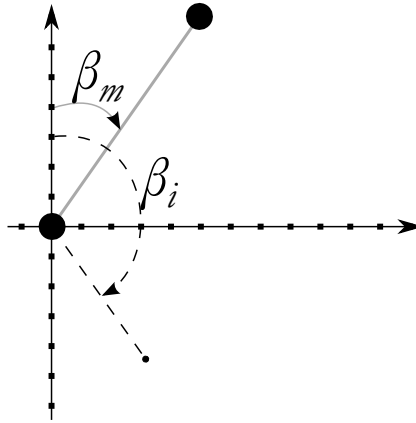


Figura 42: Ângulos entre e_i e $e_m - c_{eLocal}$

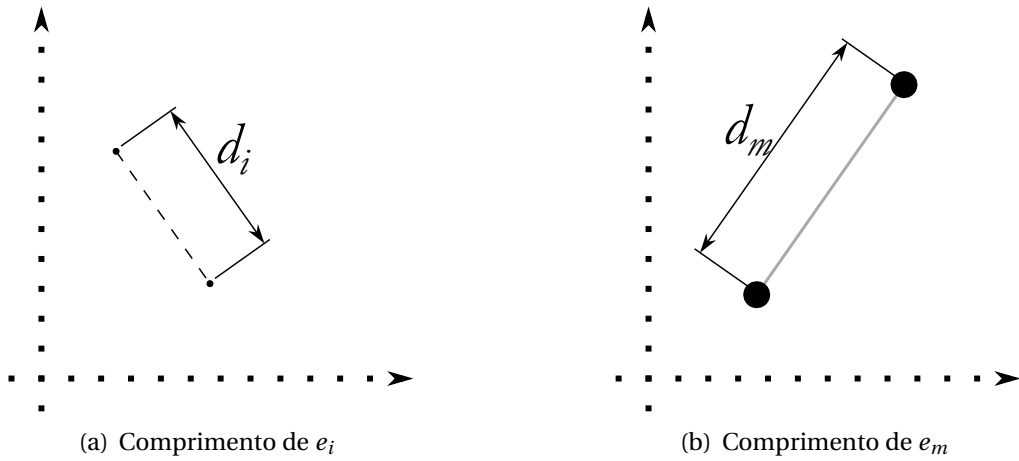


Figura 43: Comprimentos das arestas - c_{eLocal}

$$f_{eLinearLocal}(e_i, e_m) = \frac{|d_i - d_m|}{Max(d_i, d_m)} \quad (4.11)$$

Dessa forma, tomando os resultados obtidos na Eq. 4.10 e na Eq. 4.11 o cálculo do critério de avaliação de arestas local é expresso no intervalo $[0 \dots 1] \subset \mathbb{R}$, descrito pela Equação 4.12:

$$c_{eLocal}(e_i, e_m) = \frac{[(1 + f_{eLinearLocal}(e_i, e_m)) \times (1 + f_{eAngularLocal}(e_i, e_m)) - 1]}{3} \quad (4.12)$$

4.3.3.2 Critério global para avaliação de arestas - $c_{eGlobal}$

Nós desenvolvemos dois tipos de avaliação que podem ser aplicadas como critério de avaliação global na comparação de arestas.

Critério de avaliação global 01 - ($c_{eGlobal_01}$): leva em consideração a disposição das arestas em relação à estrutura formada pela descida em profundidade apresentada na seção 4.3.2 através de comparações entre os pontos médios de cada aresta existente nas subárvores formadas durante a comparação em níveis.

Esse critério de avaliação leva em conta a estrutura do grafo, tomando como base uma comparação dos ângulos formados entre a aresta ligada diretamente ao vértice comparado (adjacente ao vértice) e o segmento de reta formado pelos pontos médios de cada aresta em cada nível, bem como a distância entre os mesmos.

A Figura 44(a) mostra um grafo usado como exemplo para a aquisição de informações. Tendo como base o vértice v_{i0} , para o primeiro nível, os dados são tomados conforme a Figura 44(b) onde são mensurados os ângulos entre as arestas e as distâncias relativas aos pontos médios de cada uma.

A Figura 44(c) esboça o mesmo processo de avaliação para o segundo nível e analogamente são calculadas as medidas para os demais níveis.

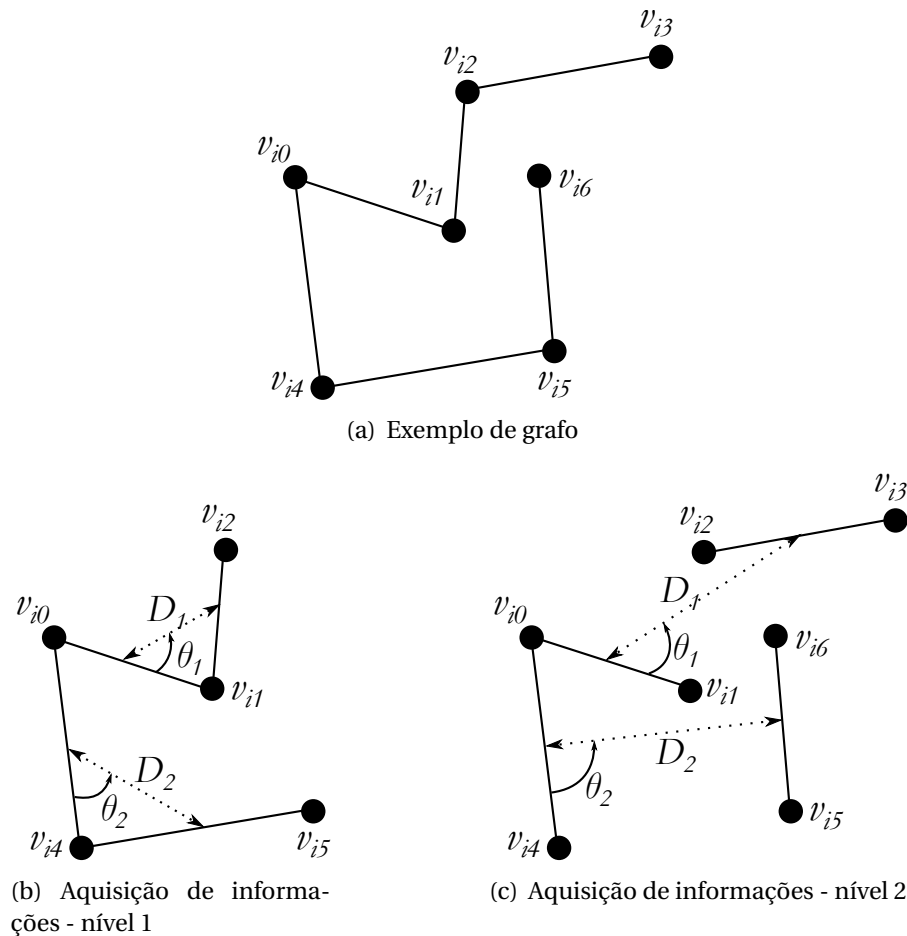


Figura 44: Aquisição de informações globais em níveis

A cada nível da árvore a avaliação global é feita com um cálculo análogo aos utilizados para o critério de avaliação local. Dessa forma, sendo θ_{ik} o ângulo obtido na tomada de dados em G_i no nível k , θ_{mk} o ângulo obtido na tomada de dados em G_m para o mesmo nível, D_{ik} a distância entre as arestas de G_i no mesmo nível k e D_{mk} a distância entre as arestas de G_m , os fatores angular e linear global são dados, respectivamente, nas Equações 4.13 e 4.14.

$$f_{eAngularGlobal_{01}}(e_i, e_m) = \begin{cases} \frac{|\theta_{ik} - \theta_{mk}|}{\text{Max}(\theta_{ik}, \theta_{mk})} & , \quad (\theta_{ik} + \theta_{mk}) \neq 0 \\ 0 & , \quad (\theta_{ik} + \theta_{mk}) = 0 \end{cases} \quad (4.13)$$

$$f_{eLinearGlobal_{01}}(e_i, e_m) = \frac{|D_{ik} - D_{mk}|}{\text{Max}(D_{ik}, D_{mk})} \quad (4.14)$$

Dessa forma, tomando os resultados obtidos nas equações 4.13 e 4.14, o cálculo do critério de avaliação de arestas global 01 ($c_{eGlobal_{01}}$) é expresso, no intervalo $[0 \dots 1]$, pela fórmula:

$$c_{eGlobal_{01}}(e_i, e_m) = \frac{[(1 + f_{eLinearGlobal_{01}}(e_i, e_m)) \times (1 + f_{eAngularGlobal_{01}}(e_i, e_m)) - 1]}{3} \quad (4.15)$$

A Figura 45 mostra um exemplo dos dados que serão comparados para uma composição de apenas um nível entre dois vértices v_{i0} de G_i e v_{m0} de G_m , respectivamente.

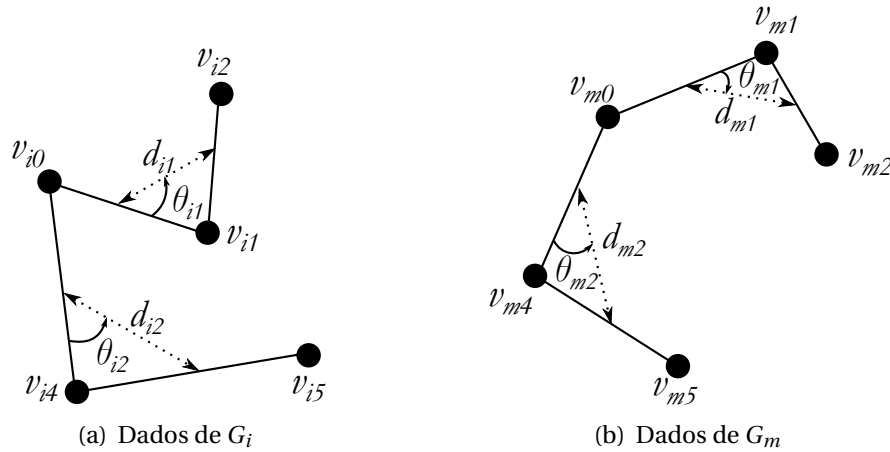


Figura 45: Informações globais para comparação no nível 1

Critério de avaliação global 02 - ($c_{eGlobal_{02}}$): para o segundo tipo de avaliação global também usaremos a comparação em níveis descrita na seção 4.3.2 diferenciando em relação

ao $c_{eGlobal_{01}}$ por não trabalhar com um ponto fixo nas arestas que estão diretamente ligadas ao vértice usado no momento da comparação, ou seja, duas arestas quaisquer podem ser comparadas diretamente através dos pontos médios de cada uma sem se atrelar a um ponto fixo em relação às arestas diretamente ligadas (adjacentes) ao vértice comparado momento.

Esse critério de avaliação global é baseado na comparação direta entre duas arestas através de seus pontos médios em cada nível. Para tanto, da mesma forma que no $c_{eGlobal_{01}}$, observamos dois fatores preponderantes, o fator angular e o fator linear.

Primeiro, calculamos o ângulo em relação à horizontal formado por esses dois pontos. Consideramos neste trabalho que esse ângulo pode variar entre 0 e $\frac{\pi}{2}$ radianos.

A Figura 46 mostra como esse ângulo é medido dos pontos médios de duas arestas comparadas onde, no eixo horizontal no plano cartesiano o ângulo tem valor 0 (zero). Esse ângulo cresce até um valor máximo de $\frac{\pi}{2}$ tanto para cima quanto para baixo não importando se está à direita ou à esquerda do eixo vertical. Dessa forma podemos observar em que pontos são obtidos os valores máximo e mínimo para o mesmo.

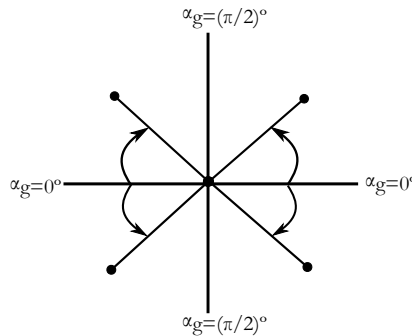


Figura 46: Valores possíveis para o fator angular global 02

Na Figura 47 observamos um exemplo de como é obtido o ângulo entre duas arestas e_i , da entrada, e e_m do modelo, a partir dos seus pontos médios.

Após a obtenção desse ângulo, o mesmo é dividido por $\frac{\pi}{2}$, que é o maior valor possível, mantendo assim o padrão de valores estipulado para este trabalho. Dessa forma temos que o fator angular global 02 é dado pela Equação 4.16:

$$f_{eAngularGlobal_{02}} = \frac{2\alpha_g}{\pi} \quad (4.16)$$

O fator linear é obtido, também, através dos pontos médios de cada aresta comparada. Como este se refere à distância entre as arestas, o mesmo é dividido pela máxima distância possível de ser obtida na imagem e, como as imagens possuem as mesmas dimensões, o fa-

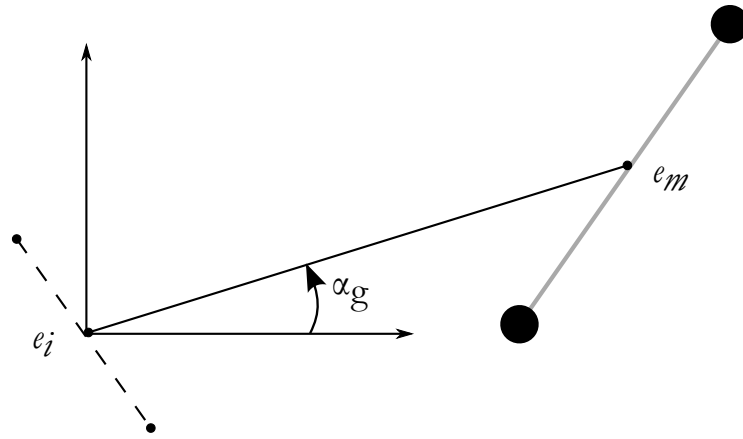


Figura 47: Tomada de fator angular global 02

tor é dividido pelo comprimento da diagonal da imagem, ou seja, a maior distância possível de se obter na imagem. Um exemplo pode ser observado na Figura 48.

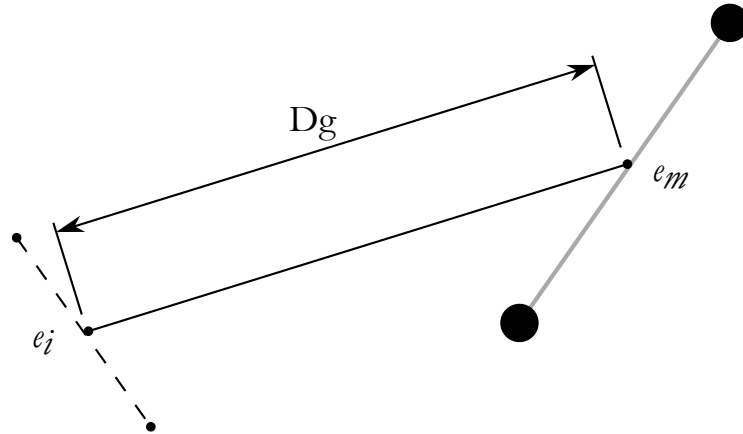


Figura 48: Tomada de fator linear global 02

Assim, para duas arestas que possuem a distância D_g entre seus pontos médios, temos que o fator linear global 02 é dado pela Equação 4.17:

$$f_{eLinearGlobal_{02}} = \frac{D_g}{MaximaDistancia} \quad (4.17)$$

Dessa forma, o critério de avaliação global de arestas 02 ($c_{eGlobal_{02}}$) é dado pela Equação 4.18:

$$c_{eGlobal_{02}}(e_i, e_m) = \frac{[(1 + f_{eLinearGlobal_{02}}(e_i, e_m)) \times (1 + f_{eAngularGlobal_{02}}(e_i, e_m)) - 1]}{3} \quad (4.18)$$

Cálculo do critério de avaliação global Com os cálculos das Equações 4.15 e 4.18, o critério de avaliação global ($c_{eGlobal}$) para duas arestas é dado usando o cálculo da Eq. 4.19:

$$c_{eGlobal}(e_i, e_m) = \sigma c_{eGlobal_{01}}(e_i, e_m) + (1 - \sigma) c_{eGlobal_{02}}(e_i, e_m) \quad (4.19)$$

onde o parâmetro σ assume os valores 0 ou 1, indicando que é possível o uso apenas da Equação 4.15 ou 4.18, exclusivamente.

4.3.3.3 Cálculo para o custo da avaliação multicritério de arestas - c'_e

Essa proposta foi estabelecida tomando como base ideias da técnica de otimização multiobjetivo [88] e [89], onde o que temos é a tarefa de minimizar uma função de custo constituída de outras subfunções.

Dessa forma, o custo da avaliação multicritério de arestas c'_e é dividido em duas subfunções que avaliam o custo local (c_{eLocal}) da Eq. 4.12 e o custo global ($c_{eGlobal}$) da Eq. 4.19. Assim, temos que nosso objetivo para o custo de associação entre arestas visa uma minimização de uma função formada por subfunções conforme a Eq. 4.20.

$$c'_e(e_i, e_m) = F_e(c_{eLocal}(e_i, e_m), c_{eGlobal}(e_i, e_m)) \quad , \quad F_e(x, y) = x * y \quad (4.20)$$

Como nosso objetivo é minimizar uma função que avalia o custo de associação entre dois grafos, optamos por usar os custos, local e global, como parâmetros em uma função de multiplicação, ou seja, tornando um critério como fator multiplicativo do outro. Como as subfunções utilizadas possuem imagem no intervalo $[0 \dots 1] \subset \mathbb{R}$, quanto menores os valores de entrada, mais rápido a função deve convergir para um valor mínimo.

Não obstante, como na operação de multiplicação, quando um dos termos é nulo, o resultado da operação se torna nulo, devemos atentar para o fato de que quando um critério, local ou global, é nulo e o outro não, indicando que as arestas comparadas são diferentes entre si, usando diretamente a multiplicação o resultado final da operação será nulo indicando que as arestas são idênticas, o que é falso.

Para tratarmos essa exceção, usaremos uma função condicional, dessa forma, o custo da avaliação multicritério de arestas é dado conforme a Equação 4.21.

$$c'_e(e_i, e_m) = \begin{cases} c_{eLocal}(e_i, e_m) \times c_{eGlobal}(e_i, e_m) & , \quad c_{eLocal}(e_i, e_m) \neq 0 \wedge c_{eGlobal}(e_i, e_m) \neq 0 \\ c_{eLocal}(e_i, e_m) + c_{eGlobal}(e_i, e_m) & , \quad c_{eLocal}(e_i, e_m) = 0 \vee c_{eGlobal}(e_i, e_m) = 0 \end{cases} \quad (4.21)$$

4.3.4 Contribuição das arestas - Termo quadrático

Nós formulamos a contribuição de arestas a partir das funções que foram propostas usando a comparação em níveis da Seção 4.3.2, onde é aplicada, a cada nível, a avaliação multicritério c'_e (Seção 4.3.3) entre duas arestas e_i e e_m usando a função descrita pela Equação 4.21.

4.3.4.1 Cálculo da contribuição de arestas - c_e

Para este procedimento são considerados dois conjuntos de arestas vinculados a cada um dos vértices v_i e v_m . Um conjunto se refere à descida em nível por uma aresta adjacente e o outro conjunto se refere à descida em nível pela outra aresta, à direita e à esquerda do vértice, respectivamente. Dessa forma teremos dois conjuntos $E_{i1} \subseteq E_i$ e $E_{i2} \subseteq E_i$ que são vinculados a v_i e outros dois conjuntos $E_{m1} \subseteq E_m$ e $E_{m2} \subseteq E_m$ vinculados a v_m .

Complementando a expressão teremos

$$E_{i1} = \{e_{i11}, \dots, e_{i1k}\}$$

o conjunto de arestas pertencente ao caminho formado à esquerda do vértice selecionado v_i , e

$$E_{i2} = \{e_{i21}, \dots, e_{i2k}\}$$

o conjunto das arestas à direita do vértice v_i .

De forma análoga, essa representação é usada para os conjuntos de arestas do vértice modelo v_m , sendo representados por

$$E_{m1} = \{e_{m11}, \dots, e_{m1k}\}$$

o conjunto de arestas à esquerda do vértice selecionado v_m , e

$$E_{m2} = \{e_{m21}, \dots, e_{m2k}\}$$

o conjunto das arestas à esquerda de v_m .

O custo final é dado pela média das somas dos custos de comparação multicritério entre arestas em cada nível.

O número máximo de arestas comparadas é $2N$ pois pode-se ter N arestas formando a subárvore à direita do vértice e N arestas na subárvore à esquerda.

Assim, sendo $2N$ o número máximo de arestas comparadas e c'_e o custo para associar uma aresta da entrada e outra do modelo através da avaliação multicritério, temos o custo da contribuição de arestas $c_e(v_i, v_m)$, conforme a equação 4.22, dada por:

$$c_e(v_i, v_m) = \frac{\left[\sum_{k=1}^{\min(|E_{i1}|, |E_{m1}|)} (c'_e(e_{i1k}, e_{m1k})) + \sum_{k=1}^{\min(|E_{i2}|, |E_{m2}|)} (c'_e(e_{i2k}, e_{m2k})) \right]}{\min(|E_{i1}|, |E_{m1}|) + \min(|E_{i2}|, |E_{m2}|)} \quad (4.22)$$

sendo que para um vértice isolado, ou seja, sem ligação com outros vértices, esse custo é zero.

4.4 Método de associação

Nosso algoritmo trabalha em duas fases, uma construtiva, com a geração de uma solução inicial gulosa, e uma fase de otimização em que trabalhamos com uma estratégia heurística a fim de proceder com um refinamento da solução obtida inicialmente.

4.4.1 Geração da solução inicial

Para a geração da solução inicial, usamos um algoritmo simples baseado na estratégia apresentada por [3] em sua aplicação no cálculo de homomorfismo usando através da comparação entre cada vértice do grafo de entrada e cada vértice do grafo de modelo.

Primeiro geramos todas as associações possíveis que são armazenadas em uma lista. Após isso, ordenamos a lista de associações e as menos custosas são escolhidas para a construção da solução inicial. O pseudo-código do algoritmo de construção da solução inicial é apresentado no Algoritmo 3 cujo resultado é denominado de $A_{inicial}[|V_i|][2]$ o qual indica a matriz solução explicada na seção 3.5.1.

O pseudo-código do Algoritmo 3 pode ser detalhado como a seguir:

- Linha 1: temos a quantidade de vértices em G_i .
- Linha 2: temos a quantidade de vértices em G_m .

Algoritmo 3: Geração da solução inicial

Dados: $G_i(V_i, E_i), G_m(V_m, E_m)$
Resultado: Conjunto de associações $A_{inicial}[|V_i|][2]$

```

1   $q \leftarrow |V_i|;$ 
2   $p \leftarrow |V_m|;$ 
3   $A_{geral} \leftarrow geraTodasAssociacoes();$ 
                                     /* Associações entre  $V_i$  e  $V_m$  */
4   $A_{geral} \leftarrow ordenaPorCusto(A_{geral});$ 
                                     /* Ordena por custo */
5   $total \leftarrow |V_m| \times |V_i|;$ 
                                     /* Número total de associações */
6   $x = 1;$ 
7   $c = 1;$ 
8  enquanto  $x \leq total$  faça
9      se  $naoAssociou(A_{geral}[x][0])$  então
10          $A_{inicial}[c][0] \leftarrow A_{geral}[x][0];$ 
11          $A_{inicial}[c][1] \leftarrow A_{geral}[x][1];$ 
12          $associou(A_{inicial}[c][0]);$ 
13          $c = c + 1;$ 
14     fim
15      $x = x + 1;$ 
16 fim
17 se  $q > p$  então
18     para  $x = 1$  até  $q$  faça
19         se  $naoAssociou(x)$  então
20              $A_{inicial}[x][0] = x;$ 
21              $A_{inicial}[x][1] = -1;$       /* Indica a não associação do vértice */
22         fim
23     fim
24 fim

```

- Linha 3: a função *geraTodasAssociacoes()* gera as associações possíveis entre os v_i 's e os v_m 's dos grafos de entrada e do modelo, respectivamente, e armazena essas associações em uma matriz chamada de A_{geral} que possui o mesmo esquema representando as associações como a matriz de solução já explicada na seção 3.5.1.
- Linha 4: nessa linha é executada a função *ordenaPorCusto*(A_{geral}), onde é aplicado o algoritmo *quicksort* para ordenar os pares de índices existentes na matriz de A_{geral} fazendo com que estes índices fiquem dispostos em ordem crescente de custo.
- Linha 5: temos a quantidade total de associações possível de ser obtida.
- Linha 6: inicializamos o contador de associações.
- Linha 7: inicializamos o contador de associações da solução inicial a ser retornada pelo algoritmo.

- Linha 8: laço para procurar as soluções dentro do conjunto total.
- Linha 9: aqui aparece a função *naoAssociou*($A_{geral}[x][0]$) (que também é mostrada na Linha 20) que verifica se o índice da associação x na posição 0 (índice que representa o vértice de G_i) existe na matriz $A_{inicial}$, ou seja, se esse vértice já foi inserido na matriz que representa a solução inicial.
- Linha 10: inserimos índice da primeira casa da matriz A_{geral} na posição x , na posição c da matriz $A_{inicial}$.
- Linha 11: inserimos o índice da segunda casa da matriz A_{geral} na posição x na posição c da matriz $A_{inicial}$.
- Linha 12: nessa linha temos a função *associou*($A_{inicial}[c][0]$) que marca como associado o índice da associação c na posição 0 (índice que representa o vértice de G_i), ou seja, marca como inserido na matriz de solução inicial.
- Linha 13: incrementamos o contador para as posições da matriz $A_{inicial}$.
- Linha 15: incrementamos o contador para as posições da matriz A_{geral} .
- Linha 17: verificamos se existem mais vértices em G_i do que em G_m , nesse caso existirão vértices desassociados.
- Linha 18: laço para verificar os índices dos vértices de G_i .
- Linha 19: verifica se um vértice de G_i de índice x não está associado.
- Linha 20: insere o índice do vértice de G_i na matriz $A_{inicial}$.
- Linha 21: insere o número (-1) na segunda casa da matriz $A_{inicial}$ indicando que o vértice x de G_i está desassociado.

Conforme o proposto, a indicação de um vértice do grafo modelo representado por um número negativo (posição dois da matriz de solução) denota a não associação do vértice referente à posição um, da matriz solução, no grafo de entrada.

A seguir serão descritos os algoritmos que geram as associações (Alg. 4) entre os vértices e posteriormente as ordenam (Alg. 5).

O Algoritmo 4 mostra a geração das associações entre os vértices de G_i e G_m onde são armazenadas em uma variável auxiliar chamada A_{geral} . Esse algoritmo tem os seguintes detalhes em suas linhas:

Algoritmo 4: Geração das associações entre vértices de G_i e G_m

Dados: $G_i(V_i, E_i), G_m(V_m, E_m)$
Resultado: Conjunto de associações $A_{geral}[|V_i| * |V_m|][2]$

```

1   $q \leftarrow |V_i|;$ 
2   $p \leftarrow |V_m|;$ 
3   $A_{geral} \leftarrow \emptyset;$ 
4   $contador = 0;$ 
5  para  $x = 1$  até  $q$  faça
6      para  $y = 1$  até  $p$  faça
7           $A_{geral}[contador][0] = x;$ 
8           $A_{geral}[contador][1] = y;$ 
9           $contador = contador + 1;$ 
10     fim
11 fim

```

- Linha 1: inicia uma variável com o número de vértices em G_i .
- Linha 2: inicia uma variável com o número de vértices em G_m .
- Linha 3: a matriz auxiliar A_{geral} está vazia no início do processo.
- Linha 4: inicializa um contador para as posições da matriz A_{geral} .
- Linha 5: laço para os índices dos vértices de G_i .
- Linha 6: laço para os índices dos vértices de G_m .
- Linha 7: a primeira posição de A_{geral} recebe o índice do vértice v_i .
- Linha 8: a segunda posição de A_{geral} recebe o índice do vértice v_m .
- Linha 9: o contador de posições é incrementado.

Algoritmo 5: Ordenação, através do custo, das associações entre vértices de G_i e G_m

Dados: $A_{geral}[\][\], c$

 /* c é a função de custo para o problema dada pela Eq. 4.1 */

Resultado: Associações ordenadas entre vértices de G_i e G_m ordenadas por custo

```

1   $A_{geral} \leftarrow QUICKSORT(A_{geral}, c);$ 

```

O Algoritmo 5 representa o processo de ordenação por custo dos índices na matriz auxiliar que guarda as associações de vértices entre G_i e G_m . Ele é uma máscara para usar o algoritmo QuickSort e tem como parâmetros de entrada a matriz auxiliar A_{geral} , que terá seus índices ordenados ao final do processo, e a função de custo definida na Equação 4.1.

Conforme se observa, a chamada para o procedimento QuickSort está presente na linha 1 do Algoritmo 5.

4.4.2 Procedimento de otimização

Dada a natureza do problema da associação de grafos, classificado como um problema NP-completo ([3]), um método de aprimoramento de soluções pode ser vislumbrado como um meio de aperfeiçoar as respostas das aplicações aqui desenvolvidas. Para este trabalho faremos uso da heurística *2-opt*.

A técnica *2-opt*, proposta originalmente para resolver o problema do caixeiro viajante[20], em sua concepção clássica encontrada na literatura, é descrita como uma estratégia de melhoria partindo de um ciclo hamiltoniano¹ [19]. A heurística *2-opt* é um procedimento derivado do conjunto de heurísticas *k-opt* ou de *k*-substituições [19], com o objetivo de encontrar ciclos hamiltonianos de melhor custo, trocando-se dois arcos do ciclo com dois arcos não pertencentes ao ciclo.

Em [20] e [90] é explicado o movimento *2-opt*, onde uma nova solução é gerada através da remoção de dois arcos, então os caminhos gerados são reconectados a dois novos arcos. Um movimento *2-opt* é ilustrado na Figura 49.

- para cada arco formado no ciclo (diz-se arco o conjunto de dois vértices adjacentes inclusive o último vértice e o primeiro, os quais constituem o último arco do conjunto) faz-se uma recombinação entre todos os outros arcos não adjacentes ao selecionado;
- a recombinação entre dois arcos é feita segundo a inversão de posições (a inversão se dá pela troca de seus índices, referentes à sequência) sendo o arco $\overrightarrow{12}$ comparado com o arco $\overrightarrow{45}$ e averiguada a melhora pela troca efetuada, a inversão de posições ocasiona dois novos arcos $\overrightarrow{14}$ e $\overrightarrow{25}$ (Figura 49);
- caso os dois novos arcos gerados proporcionem uma diminuição no valor do custo (diz-se custo de uma forma genérica, geralmente associada a distância entre cidades como no caso do Problema do Caixeiro Viajante) atribuído ao ciclo original, a troca (inversão de arcos) é mantida e passa-se à próxima comparação de arcos;
- caso a troca de arcos não ofereça melhora esta então é desfeita e passa-se à próxima comparação de arcos;

¹Um ciclo hamiltoniano em um grafo é um ciclo que passa apenas uma vez por todos os vértices do grafo.

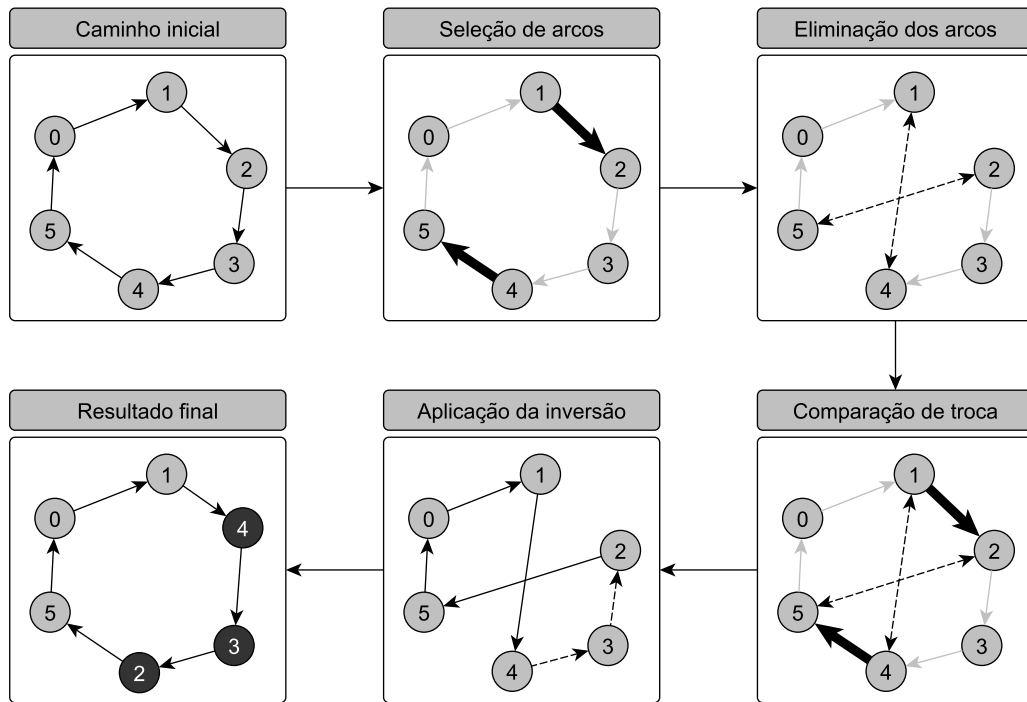


Figura 49: Exemplo de movimento *2-opt*

- o esquema *2-opt* termina assim que todos os arcos, pela sequência estabelecida para comparação, tiverem sido avaliados.

Esse procedimento de remoção de arcos e reconfiguração do caminho é repetido até que uma solução melhor seja encontrada ou que se esgotem as possíveis combinações de arcos.

4.4.2.1 Adaptação da heurística *2-opt* para melhoramento da solução do PPM

Neste trabalho, adaptamos a técnica *2-opt* original para ser utilizada na resolução do PPM. Tendo em vista que não trabalhamos com ciclos neste problema, mas com uma combinação de números, dois a dois, que representam os vértices associados de cada grafo, o que será avaliado são trocas sequenciais entre cada associação gerada. O algoritmo é bastante simples e dada a estrutura de representação do conjunto de associações, pode ser observado no pseudocódigo mostrado no Algoritmo 6, detalhado a seguir:

- Linha 1: é inicializado o contador de associações com o total de associações que foram geradas entre um grafo G_i e um grafo G_m qualquer.
- Linha 2: a variável *aprimorou*, que verifica se há melhora em um movimento *2-opt*, recebe o valor *false*.

Algoritmo 6: Adaptação de *2-opt*

Dados: Conjunto de associações $A[2]$
Resultado: Associações melhoradas $A[2]$

```

1   $n \leftarrow |A|;$ 
2   $aprimorou = false;$ 
3  para  $x = 1$  até  $n$  faça
4      para  $y = 1$  até  $n$  faça
5          se  $x \neq y$  então
6               $custoAtual \leftarrow c(A[x][0], A[x][1]) + c(A[y][0], A[y][1]);$ 
7               $custoNovo \leftarrow c(A[x][0], A[y][1]) + c(A[y][0], A[x][1]);$ 
8              se  $custoNovo < custoAtual$  então
9                   $auxiliar = A[x][1];$ 
10                  $A[x][1] = A[y][1];$ 
11                  $A[y][1] = auxiliar;$ 
12                  $aprimorou = true;$ 
13             fim
14         fim
15     fim
16 fim
17 se  $aprimorou$  então
18      $2optAdaptado$ 
19 fim

```

- Linha 3: inicia o primeiro laço que seleciona uma associação a ser comparada.
- Linha 4: inicia o segundo laço que seleciona outra associação a ser comparada.
- Linha 5: verifica se as associações são diferentes.
- Linhas 6 e 7: se as associações são diferentes então são calculados o novo custo e o custo atual para comparação.
- Linha 8: verifica se o novo custo é menor que o atual.
- Linhas 9 a 11: troca de índices na solução atual.
- Linha 12: marca a variável *aprimorou* como verdadeira.
- Linha 17: verifica se houve melhora no procedimento.
- Linha 18: executa novamente o procedimento *2-opt*.

A Figura 50 ilustra um exemplo da adaptação proposta. É selecionada uma associação $a_x = (v_{ix}, v_{mx})$ do conjunto de associações S , formada por dois números que representam um

vértice de G_i e um de G_m , respectivamente. Comparamos os custos de troca com todas as outras associações $a_y = (v_{iy}, v_{my}) \in S$.

Essa comparação é efetuada tomando a soma $c_1 + c_2$ dos custos atribuídos às associações no momento corrente, sendo $c_1 = c(v_{ix}, v_{mx})$ e $c_2 = c(v_{iy}, v_{my})$. Em seguida calcula-se dois novos custos $c'_1 = c(v_{ix}, v_{my})$ e $c'_2 = c(v_{iy}, v_{mx})$. Caso $(c'_1 + c'_2) < (c_1 + c_2)$ a troca é efetuada e é selecionada outra associação da solução corrente para proceder com a comparação. Caso não ocorra melhoria, as associações verificadas permanecem inalteradas e outra associação da solução corrente é selecionada.

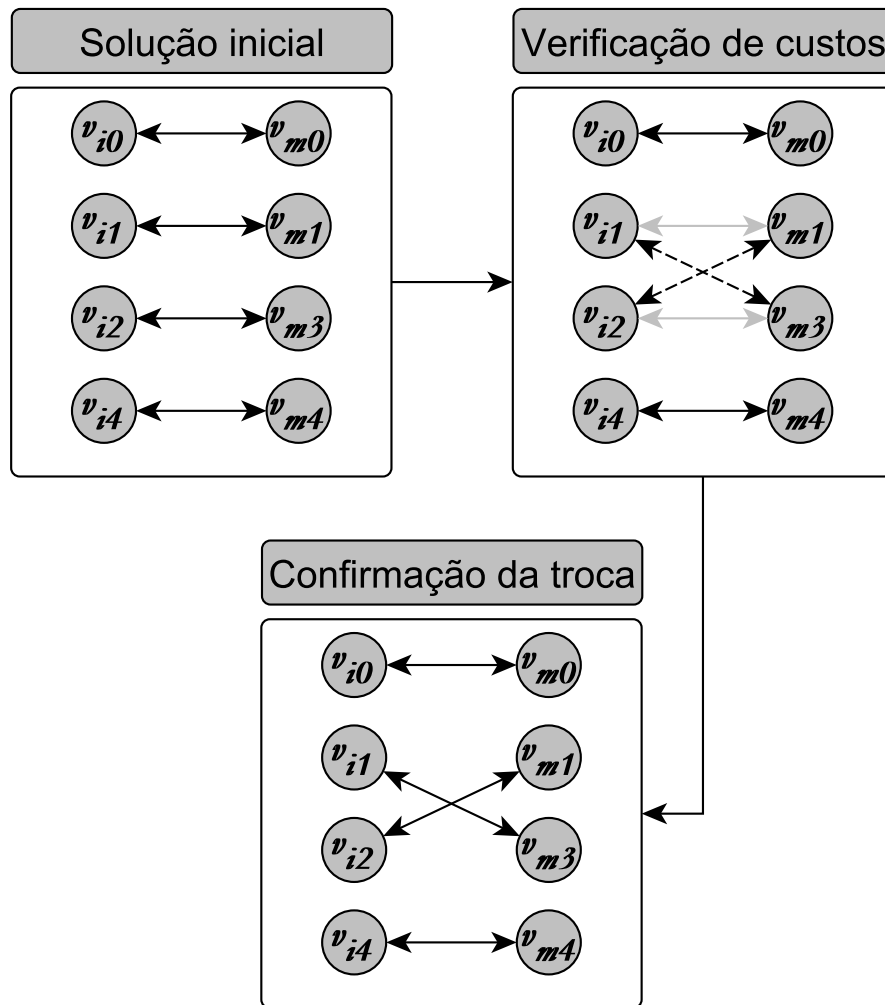


Figura 50: Exemplo da adaptação 2-opt para nosso trabalho

Para garantir como resultado do procedimento, um casamento um para um, é efetuado um teste de desempate em que, para cada vértice v_m do modelo que aparece repetido na matriz solução são verificadas todas as associações vinculadas a ele, sendo eliminadas as de pior custo, preservando apenas a melhor entre as associações para cada v_m .

4.4.3 Otimização para o tempo de execução

Através dos métodos propostos nesse capítulo, observamos ser possível, em alguns casos, usar uma estratégia para acelerar o processo que avalia a associação entre dois conjuntos de pontos.

Essa estratégia analisa a sobreposição de dois grafos G_i e G_m em um único plano e utiliza uma variável, que chamamos de distância mínima de associação (d_{ma}), para averiguar a possibilidade de associar um vértice v_i a um vértice v_m .

O procedimento pode ser explicado tomando como exemplo a Figura 51(a), a qual exibe um grafo de entrada G_i , e a Figura 51(b), que ilustra um grafo do modelo G_m . Tomando o vértice representado por v_{i2} como candidato a se associar a um vértice de G_m , procedemos com a seguinte avaliação:

- Através da sobreposição dos grafos, são selecionados os vértices em G_m que estão dentro do círculo de raio igual à d_{ma} .
- Apenas os vértices que satisfazem essa condição podem ser associados ao vértice v_{i2} .

O procedimento é ilustrado na Figura 52 onde observamos que apenas os vértices v_{m1} e v_{m2} de G_m estão aptos a serem avaliados para associação com v_{i2} . Com isso, os custos de associação entre v_{i2} e os demais vértices de G_m não são calculados, o que reduz o tempo de execução do algoritmo.

Esse processo de seleção de vértices através da distância mínima de associação pode facilmente ser empregado no código do Algoritmo 4, com a inclusão de um condicional entre as linhas 7 e 8, através de uma simples verificação de distância euclidiana entre os vértices.

Dessa forma o código do processo de geração das associações entre os vértices de G_i e G_m , com esse processo de verificação, fica como o exposto no Algoritmo 7.

Essa técnica é útil quando o procedimento de classificação trabalhar com grandes massas de dados, ou seja, no caso de um conjunto de protótipos com elevado número de imagens. Ela pode ser usada em casos onde não ocorram rotações muito perceptíveis entre as imagens de entrada e do modelo.

A execução do algoritmo proposto para avaliar o *matching* entre G_i e G_m pode ser representada através do fluxograma mostrado na Figura 53.

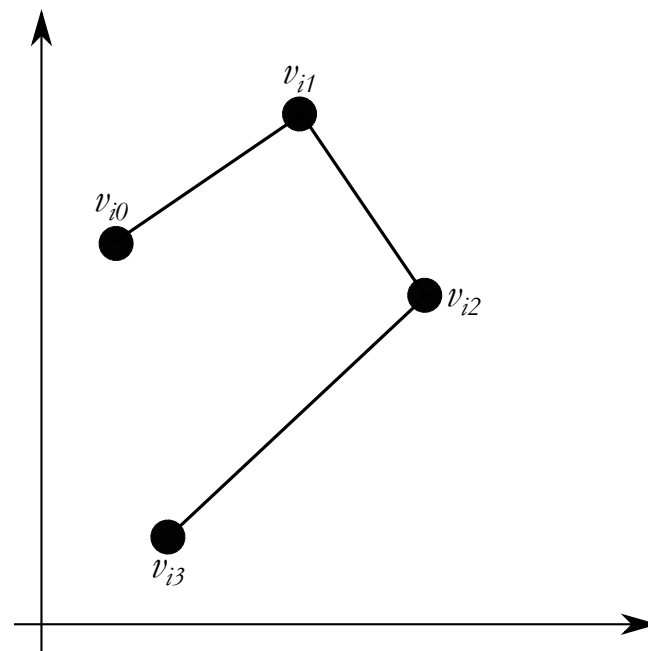
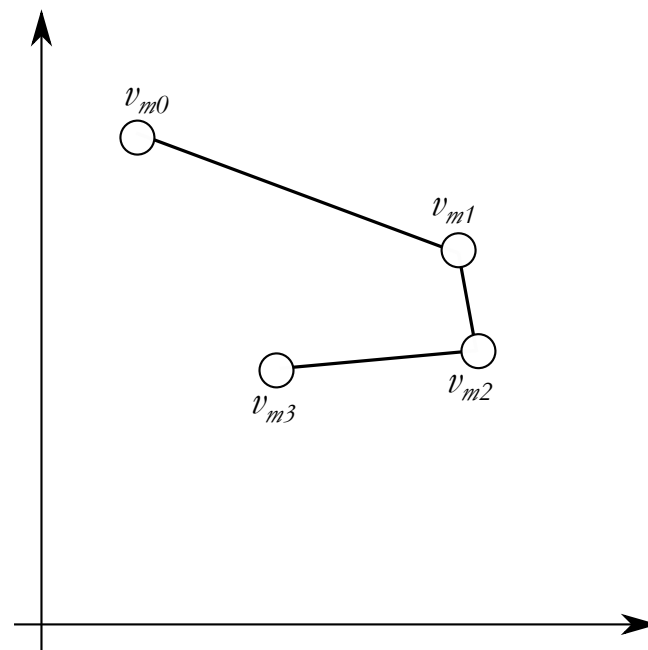
(a) G_i (b) G_m

Figura 51: Grafos de entrada e do modelo

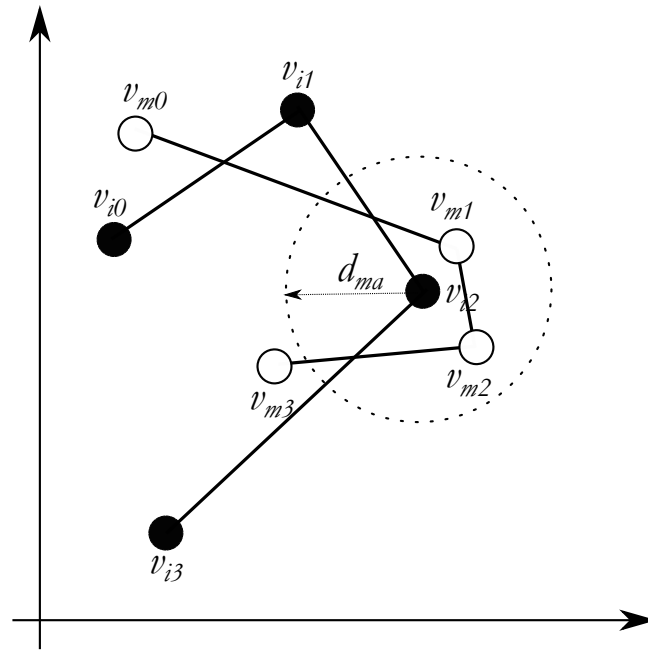


Figura 52: Avaliação da distância mínima de associação

4.4.4 Geração da solução final - melhor associação entre grafos

A associação final é dada através do Algoritmo 8 que recebe como parâmetros de execução um grafo de entrada G_i e um conjunto de protótipos (grafos modelo) B para averiguar qual grafo pertencente à B produz a melhor associação com o grafo G_i .

A explicação linha a linha do Algoritmo 8 é dada a seguir:

- Linha 1: a variável p recebe como valor o número o total de grafos do modelo.
- Linha 2: a variável que controla o menor custo vale inicialmente infinito (ou um valor muito grande).
- Linha 3: a variável que indica qual grafo do banco de modelos é melhor recebe o valor -1, indicando que não há escolha feita.
- Linha 4: laço para caminhar na lista de grafos de modelo.
- Linha 5: um grafo auxiliar recebe os dados do grafo modelo na posição x .
- Linha 6: a variável s recebe a solução inicial gerada pelo Algoritmo 3.
- Linha 7: é aplicada sobre a variável s a heurística de melhoramento *2-opt*.
- Linha 8: é eliminada a duplicidade de vértices do grafo de entrada na solução corrente.

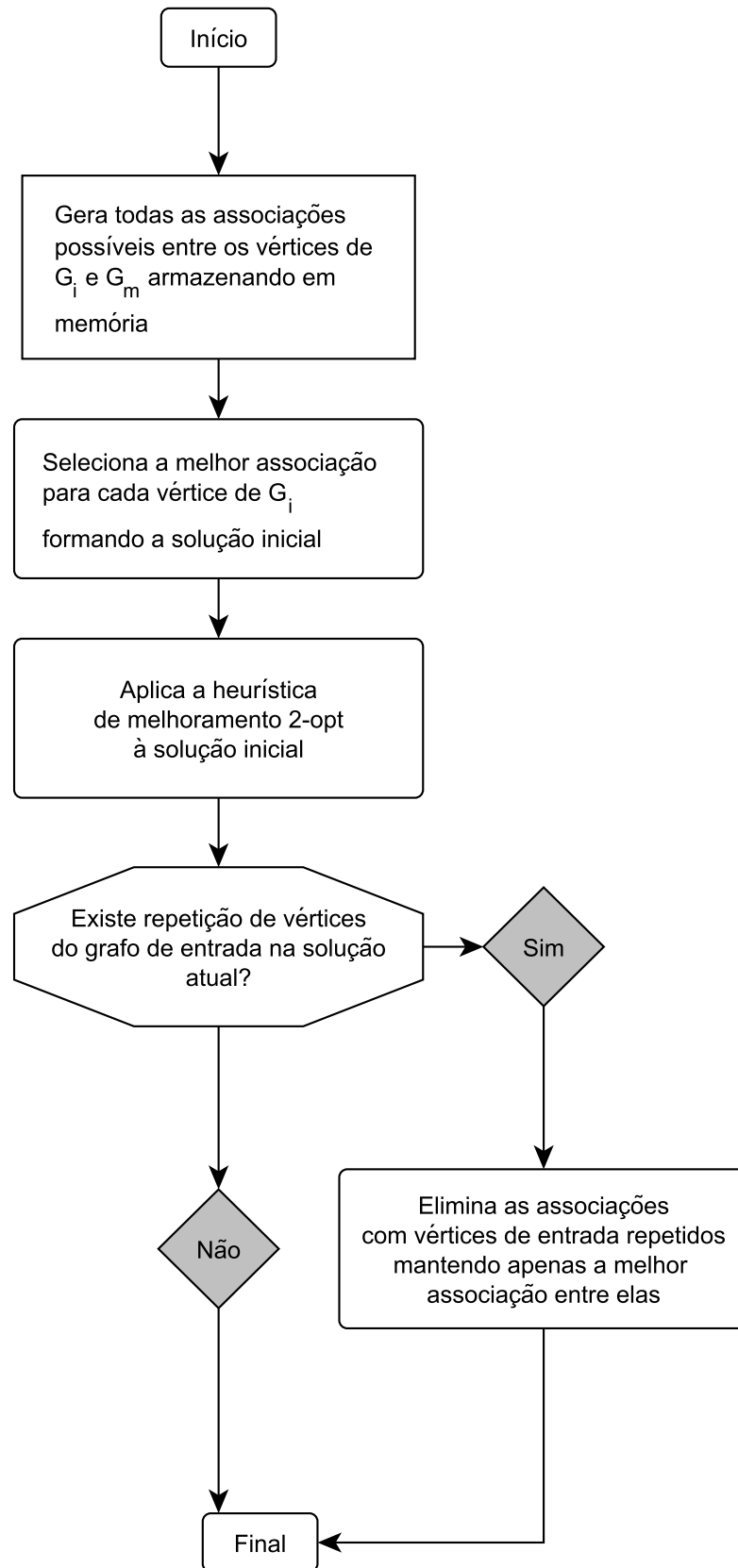


Figura 53: Fluxograma representando o esquema de geração das associações entre vértices de G_i e G_m

Algoritmo 7: Geração das associações entre vértices de G_i e G_m com comparação de distância euclidiana entre v_i e v_m

Dados: $G_i(V_i, E_i), G_m(V_m, E_m)$

Resultado: Conjunto de associações $A_{geral}[|V_i| * |V_m|][2]$

```

1   $q \leftarrow |V_i|;$ 
2   $p \leftarrow |V_m|;$ 
3   $A_{geral} \leftarrow \emptyset;$ 
4   $contador = 0;$ 
5  para  $x = 1$  até  $q$  faça
6      para  $y = 1$  até  $p$  faça
7          se  $distanciaEuclidiana(v_i, v_m) \leq d_{ma}$  então
8               $A_{geral}[contador][0] = x;$ 
9               $A_{geral}[contador][1] = y;$ 
10              $contador = contador + 1;$ 
11         fim
12     fim
13 fim

```

- Linha 9: a variável c recebe o custo da solução atual.
- Linha 10: verifica se o custo atual c é maior que a variável que armazena o menor custo.
- Linha 11: se o custo atual é menor então atualiza a variável de menor custo.
- Linha 12: atualiza o índice que indica o grafo modelo que produz a melhor associação.
- Linha 15: retorna a associação entre G_i e o grafo modelo na posição específica do banco de modelos.

4.5 O PPM definido sobre os grafos de linha de G_i e G_m

Um dos desafios enfrentados no cálculo da contribuição de vértices está na correta aquisição dos dados referentes à disposição dos pontos no preenchimento dos histogramas formados pelo gráfico log-polar na técnica *Shape context*.

Em nosso trabalho construímos uma técnica que chamamos *Shape Context* para Arestas - SCA_V - a qual diferencia-se da técnica original no sentido de crescimento dos bins presentes no gráfico log-polar. Esse sentido de crescimento é dado como perpendicular à reta formada pelo traçado de uma aresta e a origem do sistema do gráfico é coincidente ao ponto médio da aresta em que se deseja fazer a aquisição dos dados.

Observamos que a técnica SCA_V se mostra limitada quando aplicadas em grafos que

Algoritmo 8: Avaliação do Match

Dados: $G_i(V_i, E_i), B[]$ /* Conjunto de grafos de modelo */

```

1  $p \leftarrow |B|$ ;
2  $custoTotalMenor = \infty$ ;
3  $indiceMenor = -1$ ;
4 para  $x = 1$  até  $p$  faça
5    $G_m \leftarrow B[x]$ ;
6    $s \leftarrow geraSolucaoInicial(G_i, G_m)$ ;
7    $2optAdaptado(s)$ ;
8    $s \leftarrow eliminaDuplicidade_{v_i}(s)$ ;
9    $c = custo(s)$ ;
10  se  $custoMenor > c$  então
11     $custoMenor = c$ ;
12     $indiceMenor = x$ ;
13  fim
14 fim
15 retorna  $associacao(G_i, B[indiceMenor])$ 

```

possuem muitos vértices isolados, pois neste caso, como não há arestas incidentes a esses vértices, não haverá também para eles, informação calculada por esta técnica.

Observando essa limitação propomos trabalhar com os grafos de linha referentes aos grafos originais, sendo $L(G_i)$ o grafo de linha do grafo G_i e $L(G_m)$ o grafo de linha do grafo G_m .

A definição de um grafo de linha diz que: dado um grafo $G(V, E)$, o seu grafo de linha $L(G)$ é um grafo sobre o conjunto de arestas E em que, tendo x e y um par de vértices de $L(G)$, $x, y \in E$ e x, y são vértices adjacentes se e somente se x, y são arestas adjacentes em G [91].

Com isso visamos obter as seguintes vantagens:

- Eliminaríamos vértices isolados em nosso trabalho, os quais não geram SCA_V .
- As arestas dos grafos originais serão tratadas como vértices nos grafos de linha e terão um sentido de aquisição de informações proporcional à possíveis rotações do grafo original.
- O casamento dos grafos G_i e G_m é avaliado como uma associação de pontos sobre os seus respectivos grafos de linha, o que não altera as características dos algoritmos propostos.

4.5.1 Aquisição de informações no grafo de linha

Na Figura 54, mostramos um exemplo de um grafo e seu grafo de linha correspondente. O grafo G formado pelos vértices (A,B,C,D,E) da Figura 54(a) tem seu grafo de Linha $L(G)$ formado pelos vértices (1,2,3,4), que representam as arestas de G , representado na Figura 54(b).

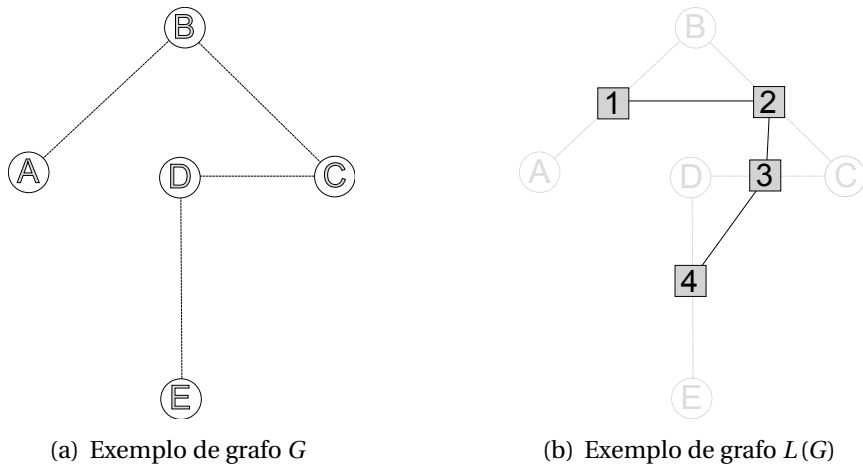


Figura 54: Um grafo G e seu correspondente $L(G)$

A aquisição de pontos para o preenchimento dos histogramas para o vértice 1 de $L(G)$ é mostrada na Figura 55. Nela podemos observar que o sentido dos eixos do gráfico log-polar foi rotacionado fazendo com que o eixo vertical desse gráfico esteja perpendicular à aresta referente no grafo G .

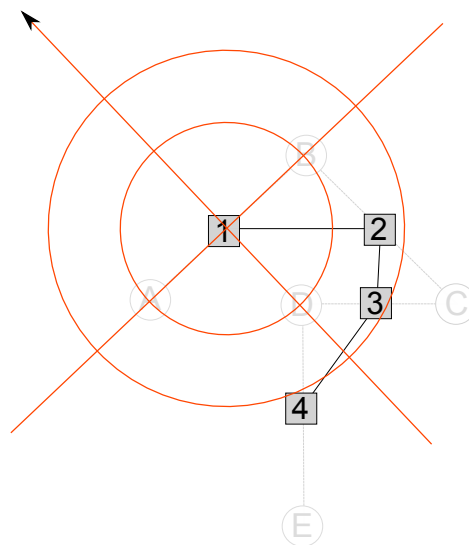


Figura 55: SCA para o vértice 1 em $L(G)$

Em nosso trabalho lidamos com arestas não direcionadas, contudo, essa técnica está

- De posse dos cálculos de SCA_1 e SCA_2 , verifica-se qual é menos custoso e assume-se esse custo como o custo de associação entre os vértices vL_i e vL_m .

5 *Testes Computacionais*

Apresentamos neste capítulo, os resultados computacionais obtidos a partir das implementações dos métodos propostos no Capítulo 4.

5.1 Instâncias de testes - Bancos de imagens

Em nosso trabalho implementamos um algoritmo que efetua o casamento entre grafos e para constatar a eficiência desse algoritmo aplicamos testes sobre três bases de imagens específicas com características distintas e com o objetivo comum de proceder com o reconhecimento de formas.

Esse reconhecimento se dá através da aquisição das características das imagens em relação ao seu contorno com o uso de um detector Canny [34] para a utilização dessas características como informações de vértices e arestas.

As abordagens para todos os bancos de imagens utilizam a modelagem descrita na Seção 3.5.1, cujo objetivo é minimizar o custo dado pela Equação 3.4.

5.1.1 KIMIA e MPEG96¹

A base de imagens KIMIA é composta de 99 silhuetas tendo 9 categorias com 11 imagens cada e se refere aos testes efetuados em [6]. A Figura 57 mostra o conjunto de imagens utilizado.

As classes de imagens nessa base possuem características próprias. Essas características dizem respeito à presença de rotações e redimensionamentos que são observações que devem ser levadas em consideração na construção do algoritmo.

¹<http://www.lems.brown.edu/vision/researchAreas/SIID/>

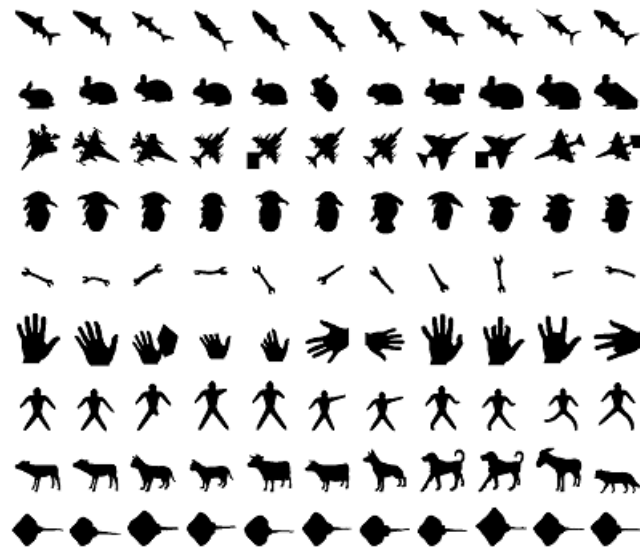


Figura 57: Banco de imagens KIMIA

Na Figura 58 observa-se um exemplo de imagens da mesma categoria mas cujo reconhecimento pode ser dificultado pelo grau de rotação de uma imagem em relação à outra. Da mesma forma, mas de maneira mais evidente, observamos o caso apresentado na Figura 59 em que a rotação é maior.

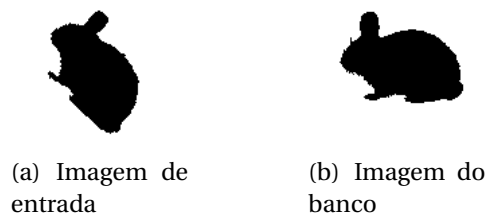


Figura 58: Dificuldades encontradas em KIMIA - Rotação

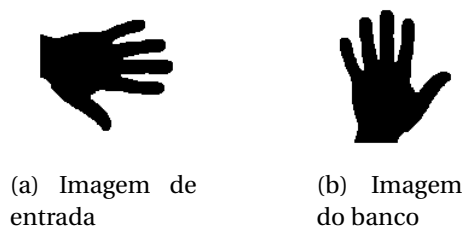


Figura 59: Dificuldades encontradas em KIMIA - Rotação

Já na situação apresentada na Figura 60 a dificuldade está a cargo do escalonamento encontrado. As duas imagens apresentadas estão inseridas na mesma classe, contudo, possuem tamanhos diferentes, o que pode ocasionar um maior esforço para efetuar uma associação entre elas.

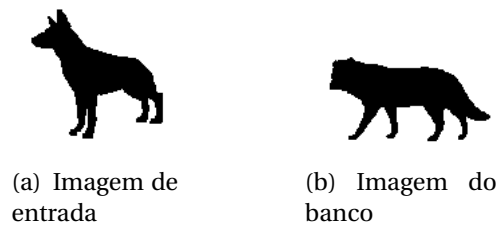


Figura 60: Dificuldades encontradas em KIMIA - Escalonamento

Em relação à base de imagens MPEG96, fazemos menção ao mesmo trabalho de [6] onde os autores usaram uma parte da base de imagens MPEG [59] selecionando 96 formas divididas em 8 categorias de 12 imagens cada.

Essa base de imagens é também formada por silhuetas e por isso foi incluída nessa seção por ter características semelhantes à base KIMIA.

A Figura 61 ilustra as imagens da base MPEG96.

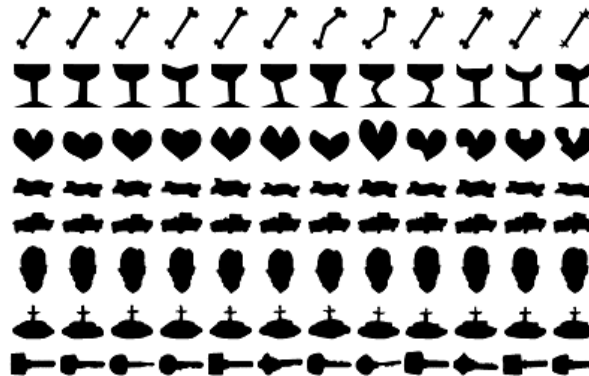


Figura 61: Imagens de MPEG. Fonte: [6]

5.1.2 COIL

O banco de imagens COIL² é referente aos trabalhos [92] e [93].

Nesse banco tem-se imagens de 20 objetos comuns, como caixas de remédio, carrinhos de brinquedo entre outros. De cada objeto foram tiradas fotografias sobre uma plataforma giratória a cada 5°.

Essa base de imagens possui um total de 1400 imagens, sendo 70 para cada objeto. A Figura 62 exemplifica como foi feito o processo de aquisição de imagens dessa base e como elas estão dispostas.

²<http://www.cs.columbia.edu/CAVE/software/softlib/coil-100.php>

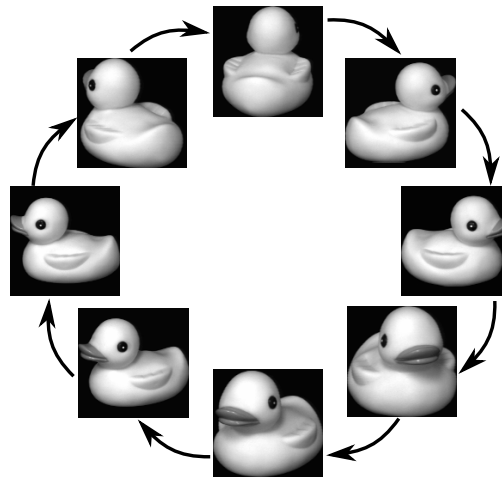


Figura 62: Tomada de imagens da base COIL

A principal vantagem observada neste banco é o fato de que suas imagens possuem as mesmas dimensões e sem rotação no plano 2D em torno do eixo Z.

5.1.3 MNIST

O banco de imagens MNIST é proveniente do trabalho de [94] e se refere à imagens de dígitos manuscritos.

Nesse banco encontramos uma massa de dados com 60000 imagens para treinamento (protótipos) e 10000 imagens de testes. Essas imagens possuem dimensões fixas de 28x28 pixels.

Não obstante, o maior obstáculo para essa implementação é a quantidade de imagens, tanto para testes quanto para treinamento.

A Figura 63 ilustra algumas imagens de dígitos que são encontrados no banco MNIST.

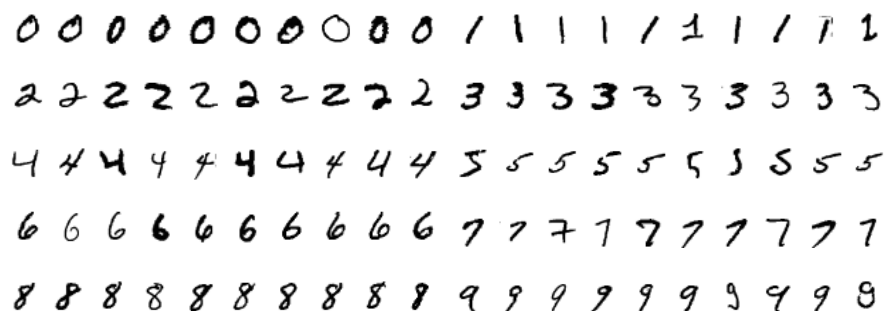


Figura 63: Imagens do banco MNIST

5.2 Testes efetuados

Nesta seção mostraremos os resultados dos testes efetuados em nosso trabalho comparando-os a resultados disponíveis na literatura.

Procedemos com testes em uma máquina equipada com um processador Core(TM)2 de 3 GHz e 2GB de memória RAM.

Os códigos foram implementados usando a linguagem JAVA e executados na plataforma Windows XP com uma JVM versão 1.7.0.

5.2.1 Testes 01 - KIMIA e MPEG96

Nesta seção nós procedemos com testes sobre a base de imagens KIMIA e MPEG96 usando o mesmo tipo de avaliação apresentado em [4, 61, 67], fazendo com que cada imagem fosse associada com o restante das imagens do banco, ou seja, geramos 98 associações para cada imagem. Essas associações, para cada imagem, são contabilizadas observando as 10 melhores associações e verifica-se ao final quantas classificações corretas foram feitas para a primeira melhor, para a segunda melhor e assim por diante até a décima melhor associação.

Nesta bateria de testes foi utilizado o Algoritmo 4 para gerar as associações entre vértices de G_i e G_m .

5.2.1.1 Parâmetros de execução

Os parâmetros usados para o Algoritmo 4 aplicado à base de imagens KIMIA e MPEG96 foram:

- $\lambda = 0,4$: aplicado na Equação 4.9.
 - O uso desse parâmetro implica em uma combinação do SCP (Eq. 4.6) e SCA_V (Eq. 4.8) aplicada na contribuição de vértices.
- $\sigma = 1$: aplicado na Equação 4.19.
 - O uso desse valor indica que o algoritmo utiliza como critério de avaliação global a função $c_{eGlobal_{01}}$ definida pela Equação 4.15.
- $N = 10$: aplicado na comparação em níveis definida na Seção 4.3.2.

- O valor desse parâmetro indica que foram usados 10 níveis de comparação para o cálculo da contribuição de arestas dado pela Equação 4.22.
- $\alpha = 0,5$: aplicado na Equação 4.1.
- Esse valor indica que a contribuição de vértices e arestas é balanceada de forma equitativa no custo de associação entre vértices de G_i e G_m .

5.2.1.2 Metodologia de teste

Nós comparamos os resultados obtidos com os encontrados em [4, 67] onde cada imagem do banco é dada como uma imagem de entrada e forma um grafo G_i . Essa imagem de entrada é então comparada a todas as outras imagens do banco formando uma listagem de classificações entre G_i e cada G_m (formado através das imagens restantes que são designadas como modelos).

Ao final desse processo, as dez melhores associações são selecionadas e compara-se a classe da imagem de entrada com as classes a que pertencem as respectivas imagens que geraram as melhores associações.

Dessa forma, ao final da execução para as 99 imagens do banco, o melhor resultado esperado é que se obtenha 99 acertos, com relação à avaliação de classes, para cada uma das dez melhores associações.

5.2.1.3 Resultados computacionais

Nós apresentamos um resumo sobre esses resultados na Tabela 2 onde denominamos a nossa técnica como “SCP+SCA+MC”. As nomenclaturas “SC”, “Shock Edit” e “IDSC+DP” indicam os resultados apresentados na literatura para *Shape Context*, *Editing Shock Graphs* e *Inner-Distance Shape Context*, respectivamente. Os trabalhos de referência são mostrados na seção 3.3 e são indicados na própria tabela.

Tabela 2: Resultados para KIMIA - 10 melhores classificações para cada imagem

Algoritmo	1º	2º	3º	4º	5º	6º	7º	8º	9º	10º
SC [4]	97	91	88	85	84	77	75	66	56	37
Shock Edit [4]	99	99	99	98	98	97	96	95	93	82
IDSC+DP [67]	99	99	99	98	98	97	97	98	94	79
SCP+SCA+MC	99	97	91	91	83	84	79	83	64	56

A Figura 64 mostra alguns exemplos de associações obtidas para a base de imagens nessa implementação.

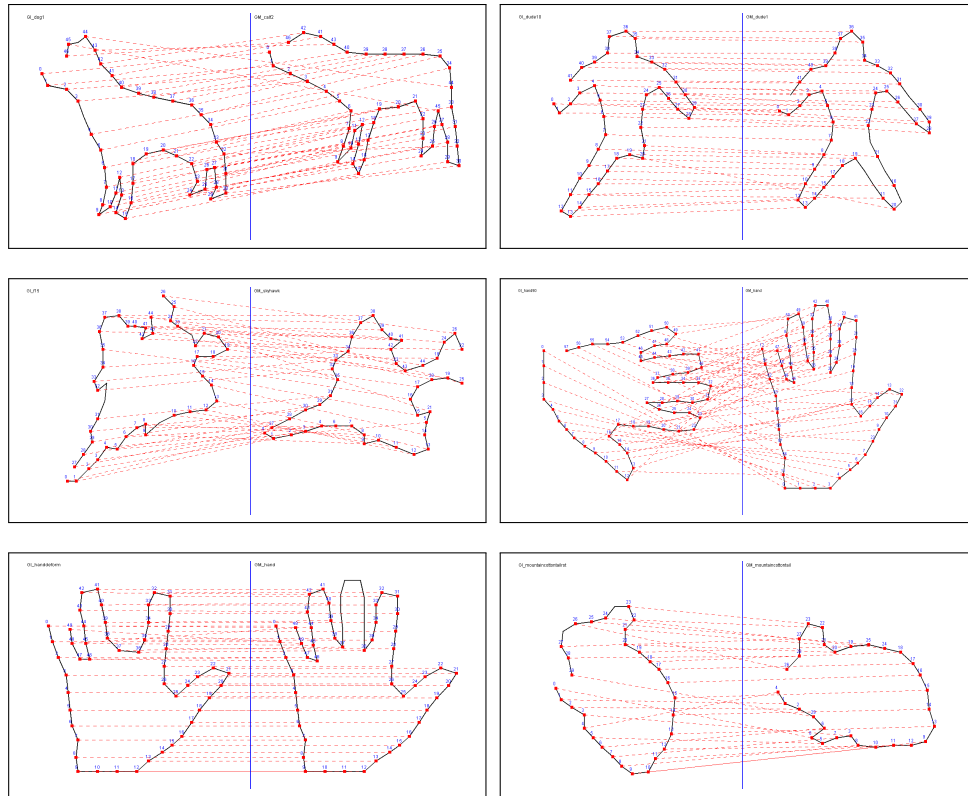


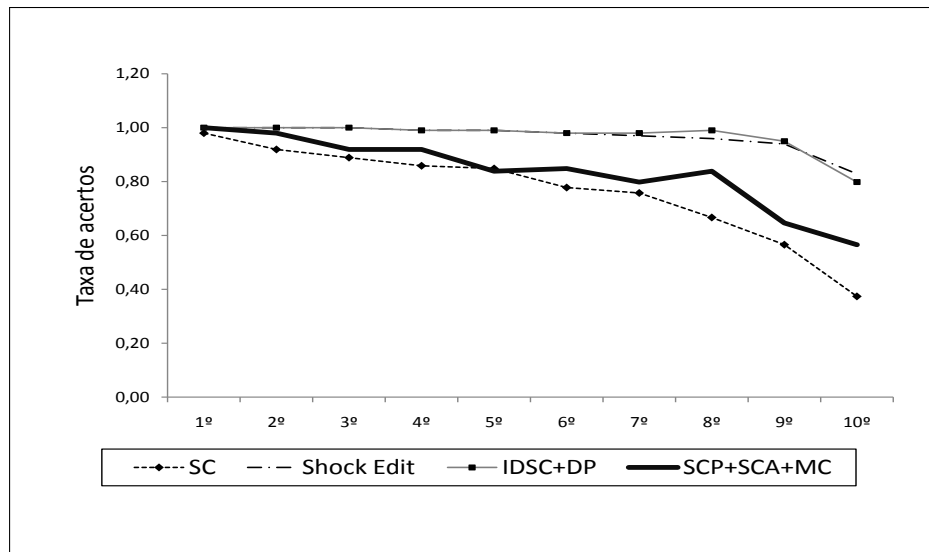
Figura 64: Associações obtidas em KIMIA

Os testes efetuados sobre a base MPEG96 procedem da mesma maneira que para a base de KIMIA apenas mudando a faixa de avaliação que aqui é de 11 melhores classificados. A Figura 61 mostra as imagens utilizadas nesse procedimento e a Tabela 3 mostra os resultados obtidos onde são comparados aos resultados da técnica *Editing Shock Graphs* [6] referenciada na tabela por Shock Edit.

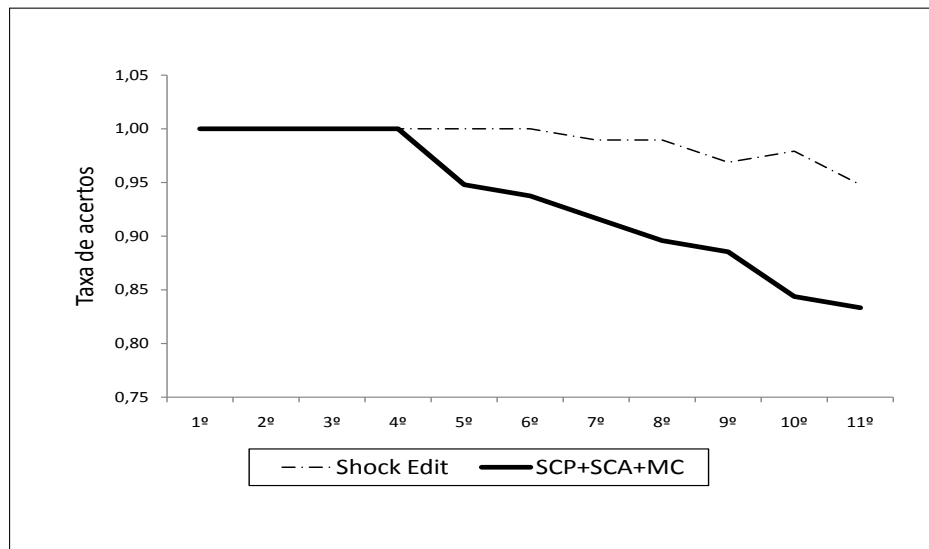
Tabela 3: Resultados para MPEG 96 - 11 melhores classificações para cada imagem

Algoritmo	1º	2º	3º	4º	5º	6º	7º	8º	9º	10º	11º
Shock Edit [6]	96	96	96	96	96	96	95	95	93	94	91
SCP+SCA+MC	96	96	96	96	91	90	88	86	85	81	80

O comportamento do algoritmo desenvolvido é observado através do gráfico mostrado nas Figuras 65(a) e 65(b), onde notamos que a eficiência do algoritmo cai muito rapidamente com o afastamento das primeiras posições para as melhores classificações obtidas, isso se deve ao fato de não se adaptar bem a algumas características das imagens como por exemplo rotações e dimensões diferentes em imagens pertencentes a uma mesma classe.



(a) Comportamento em KIMIA



(b) Comportamento em MPEG

Figura 65: Comportamento do algoritmo

5.2.2 Testes 02 - KIMIA e MPEG96

Para os testes aqui propostos executaremos o algoritmo baseado na proposta de associação utilizando grafos de linha conforme descrito na Seção 4.5. Nós iremos proceder com as mesmas bases de imagens relacionadas anteriormente de KIMIA e MPEG96.

Nós utilizamos o Algoritmo 4 para gerar as associações entre vértices de $L(G_i)$ e $L(G_m)$.

5.2.2.1 Parâmetros de execução

Os parâmetros usados para essa bateria de testes sofreram modificações para que o algoritmo proposto fosse executado utilizando a proposta sobre grafos de linha. As modificações são as seguintes:

- Para a contribuição de vértices usaremos apenas o *SCA* sobre os vértices dos grafos de linha $L(G_i)$ e $L(G_m)$ correspondentes a G_i e G_m . Ou seja, usamos apenas o *SCA* sobre cada aresta de G_i e G_m .
- Para a avaliação de arestas usamos apenas a comparação através do critério local expressa na Eq. 4.12.
- Os demais parâmetros são referentes à execução do algoritmo proposto e são os seguintes:
 - $N = 10$: para a comparação de arestas em níveis a Seção 4.3.2.
 - * Representa o uso de 10 níveis de comparação para o cálculo da contribuição de arestas dado pela Equação 4.22.
 - $\alpha = 0,6$: para a Equação 4.1.
 - * Balanceamento entre a contribuição de arestas e a contribuição de vértices de G_i e G_m .

5.2.2.2 Metodologia de teste

A metodologia de teste é a mesma apresentada na seção anterior tanto para os testes sobre a base KIMIA quanto para os testes sobre a base MPEG96.

5.2.2.3 Resultados computacionais

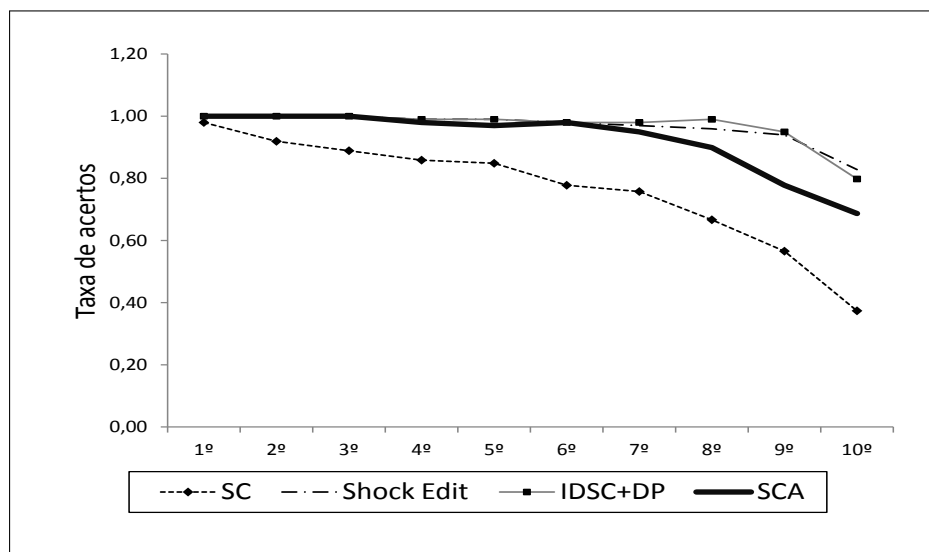
Os resultados computacionais são apresentados na Tabela 4, referente aos testes para KIMIA, e na Tabela 5 referente aos resultados para MPEG96. A nossa técnica, usando grafos de linha, é denominada nas tabelas por “SCA” e os trabalhos referentes aos demais resultados estão dispostos como referência presentes nas próprias tabelas. Os gráficos ilustrados nas Figuras 66(a) e 66(b) representam o comportamento do algoritmo aplicado às duas bases, KIMIA e MPEG96, respectivamente, com a implementação aqui proposta para grafos de linha.

Tabela 4: Resultados para KIMIA - 10 melhores classificações para cada imagem

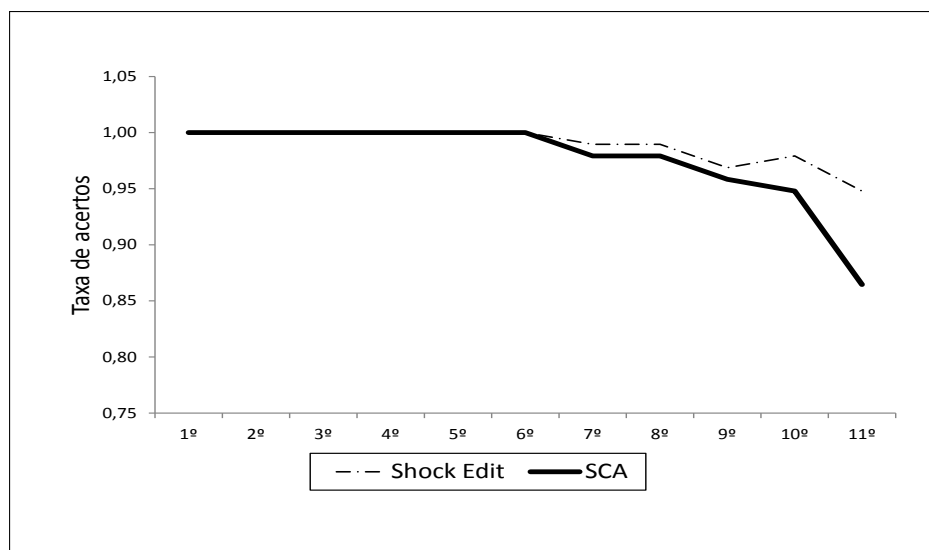
Algoritmo	1º	2º	3º	4º	5º	6º	7º	8º	9º	10º
SC [4]	97	91	88	85	84	77	75	66	56	37
Shock Edit [4]	99	99	99	98	98	97	96	95	93	82
IDSC+DP [67]	99	99	99	98	98	97	97	98	94	79
SCA	99	99	99	97	96	97	94	89	77	68

Tabela 5: Resultados para MPEG 96 - 11 melhores classificações para cada imagem

Algoritmo	1º	2º	3º	4º	5º	6º	7º	8º	9º	10º	11º
Shock Edit [6]	96	96	96	96	96	96	95	95	93	94	91
SCA	96	96	96	96	96	96	94	94	92	91	83



(a) Comportamento em KIMIA com SCA



(b) Comportamento em MPEG com SCA

Figura 66: Comportamento do algoritmo

5.2.3 Testes 03 - COIL

Para a segunda bateria de testes nós aplicamos o nosso algoritmo sobre o banco de imagens COIL [93], elaboramos um classificador 1-NN a fim de se averiguar a eficiência de nossa implementação procedendo com uma comparação com os resultados descritos em [3]. Utilizamos aqui o Algoritmo 4 para gerar as associações entre vértices de G_i e G_m .

5.2.3.1 Parâmetros de execução

Os parâmetros aplicados no algoritmo para os testes com essa base de imagens foram:

- $\lambda = 1$: aplicado na Equação 4.9.
 - O uso desse parâmetro implica no uso do *SCP* (Eq. 4.6), apenas, aplicado para a contribuição de vértices.
- $\sigma = 0$: aplicado na Equação 4.19.
 - O uso desse valor indica que o algoritmo utiliza como critério de avaliação global a função $c_{eGlobal_{02}}$ definida pela Equação 4.18.
- $N = 10$: aplicado na comparação em níveis definida na Seção 4.3.2.
 - Esse valor indica o uso de 10 níveis de comparação para o cálculo da contribuição de arestas efetuado pela Equação 4.22.
- $\alpha = 0,6$: aplicado na Equação 4.1.
 - A contribuição de vértices possui um peso ligeiramente superior à contribuição de arestas.

5.2.3.2 Metodologia de teste

Nós procedemos com os testes seguindo os passos aplicados em [3] onde o autor avalia o comportamento de seu método, que usa uma técnica baseada em propagação de crenças, através da variação do número de protótipos usado para a classificação.

Em [3], como neste trabalho, é proposto um classificador 1-NN e com base em visões igualmente espaçadas (dada a forma de aquisição de 5 em 5 graus rotacionados) das imagens foram variados os números de protótipos por classe e o comportamento da técnica foi descrita em um gráfico. Nós procedemos com esse mesmo tipo de teste.

Em um outro momento, após primeira bateria de testes, [3] utiliza uma premissa de agrupamento baseada em *k-medoids*, tendo em vista que, dada a complexidade de cada objeto, diferentes classes de objetos necessitam de quantidades diferentes de protótipos.

O algoritmo para *k-medoids* é aplicado em problemas de agrupamento de dados [95] e visa obter, dado um conjunto de dados, objetos, chamado *medoids*, pertencentes ao conjunto e que representem os grupos de dados inseridos nesse conjunto. Segundo [96] os demais objetos presentes no conjunto de dados são associados aos grupos de acordo com a semelhança com os objetos representativos - *medoids*.

Nós executamos a segunda etapa de testes usando os mesmos protótipos agrupados por [3].

5.2.3.3 Resultados computacionais

Conforme descrito na metodologia, inicialmente nós atribuímos uma variação no número de protótipos a serem usados. Usando 8 protótipos por objeto, igualmente espaçados, dessa forma obtivemos uma taxa de erro igual a 0,0338 inferior à taxa obtida em [3] que foi de 0.0581.

O gráfico presente à Figura 67 mostra o comportamento do algoritmo, através da taxa de erros, com a variação no número de visões por objeto demonstrando que o número de erros decresce com o aumento do número de protótipos, o que já era esperado.

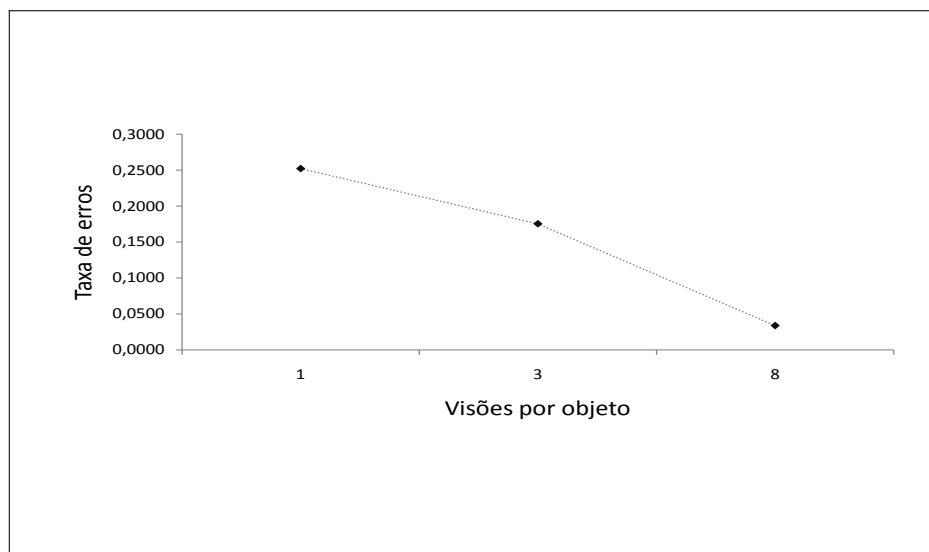


Figura 67: Taxa de erros em COIL com variação no número de protótipos

A segunda etapa de testes que diz respeito ao agrupamento *k-medoids* foi executada utilizando os mesmos 158 protótipos usados em [3] e nós obtivemos 5 erros em 1242 compara-

ções resultando em uma taxa de erro igual a 0,0040 que é inferior à taxa obtida no referido trabalho que é de 0,0161 ou 20 erros em 1242 comparações.

As Figuras 68, 69, 70, 71 e 72 mostram os erros encontrados durante os testes tendo acima as imagens do banco (de teste à esquerda e de treinamento à direita) e abaixo a imagem que mostra as associações obtidas. Os 158 protótipos utilizados são mostrados nas Figuras 73, 74 e 75.

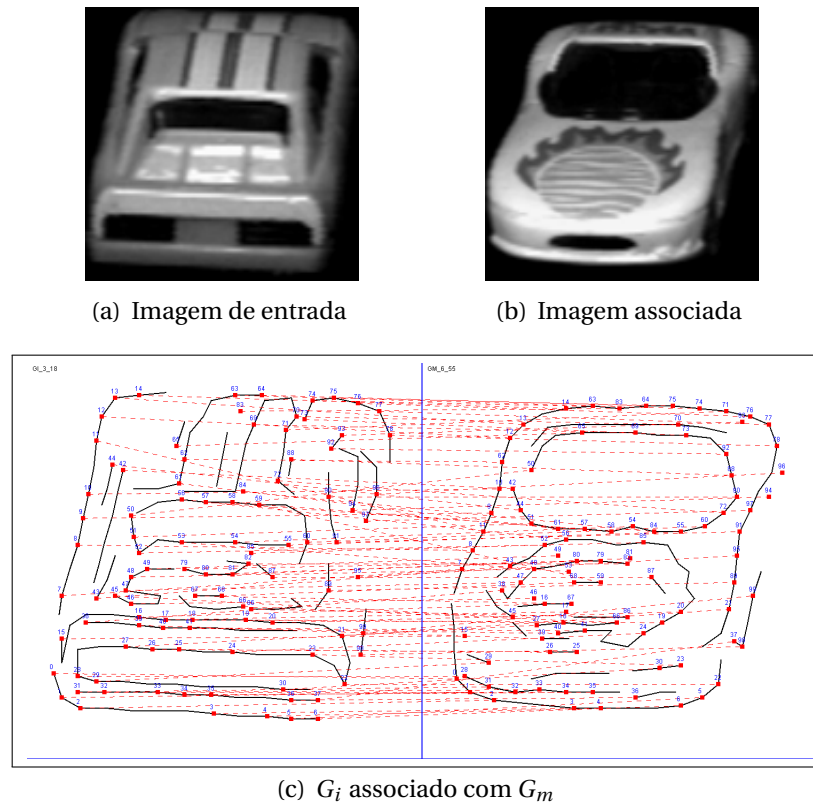
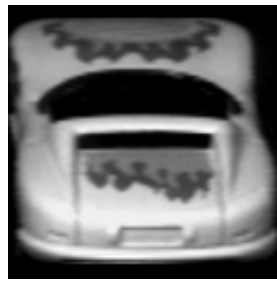
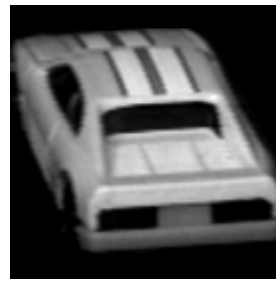


Figura 68: Erro obtido em COIL



(a) Imagem de entrada



(b) Imagem associada

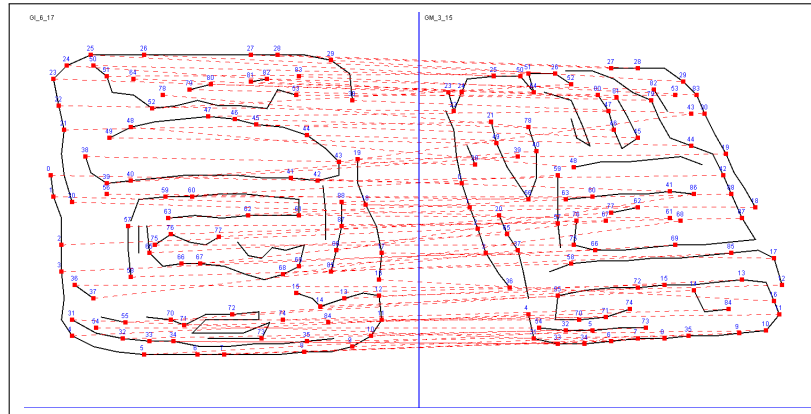
(c) G_i associado com G_m

Figura 69: Erro obtido em COIL



(a) Imagem de entrada



(b) Imagem associada

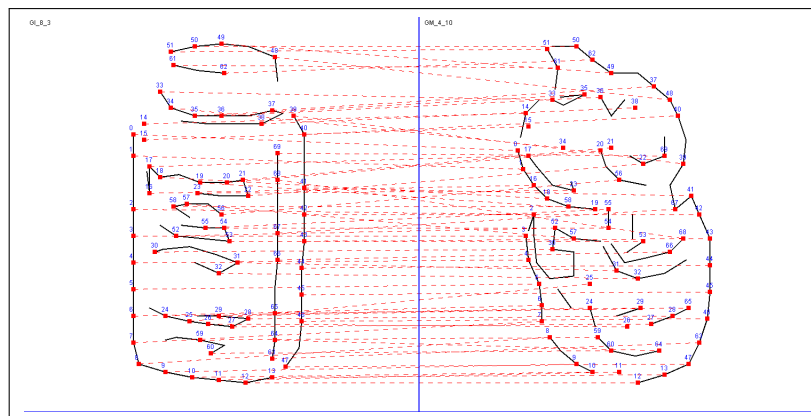
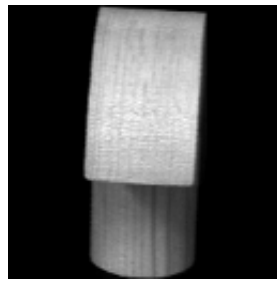
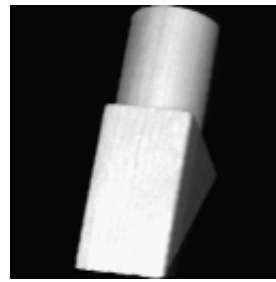
(c) G_i associado com G_m

Figura 70: Erro obtido em COIL



(a) Imagem de entrada



(b) Imagem associada

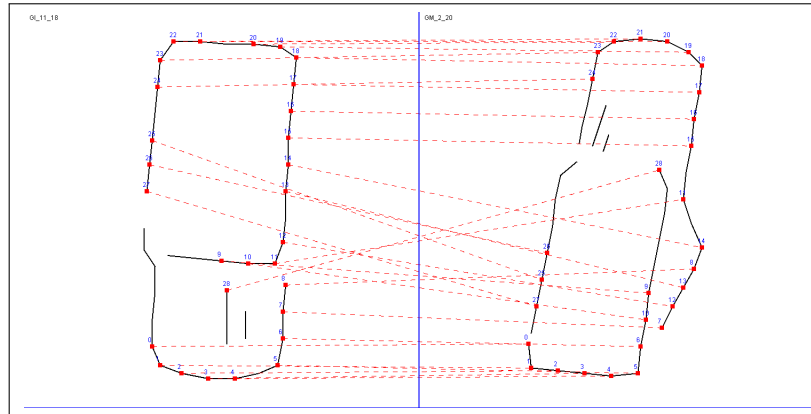
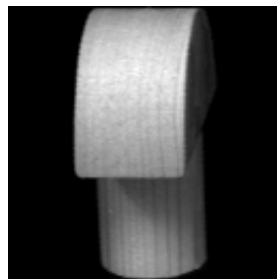
(c) G_i associado com G_m

Figura 71: Erro obtido em COIL



(a) Imagem de entrada



(b) Imagem associada

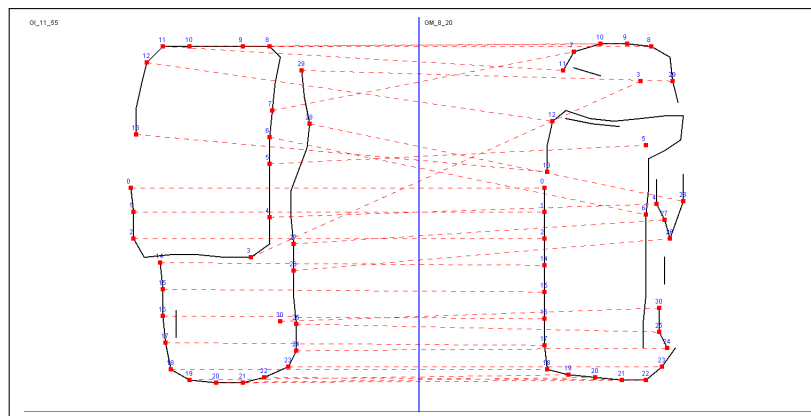
(c) G_i associado com G_m

Figura 72: Erro obtido em COIL



Figura 73: Protótipos usados em COIL

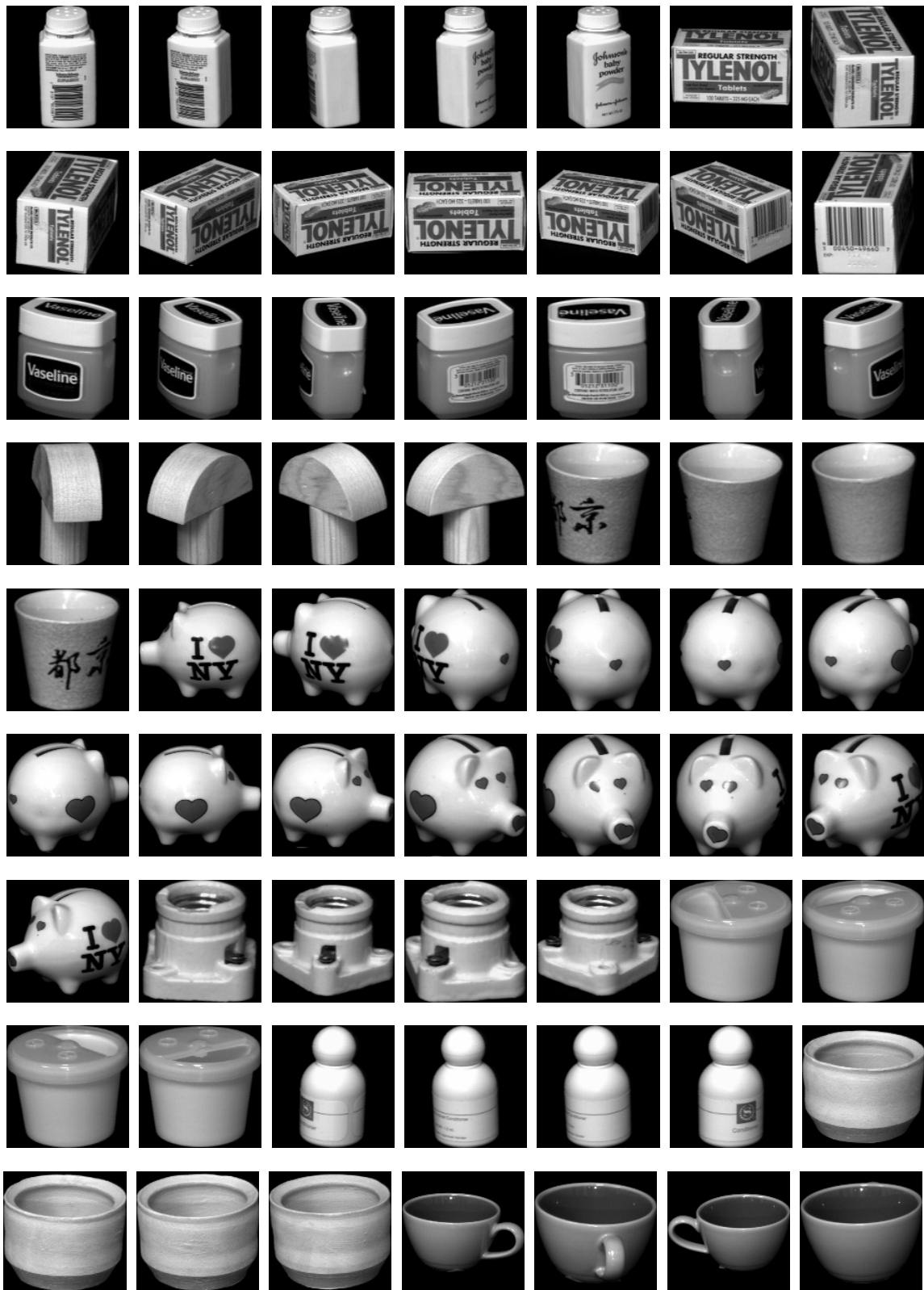


Figura 74: Protótipos usados em COIL

5.2.4 Testes 04 - MNIST

Em nossa terceira bateria de testes propusemos aplicar nosso algoritmo sobre a base de imagens MNIST. Nesta bateria de testes foi utilizado o Algoritmo 7, mostrado na Seção 4.4.3,

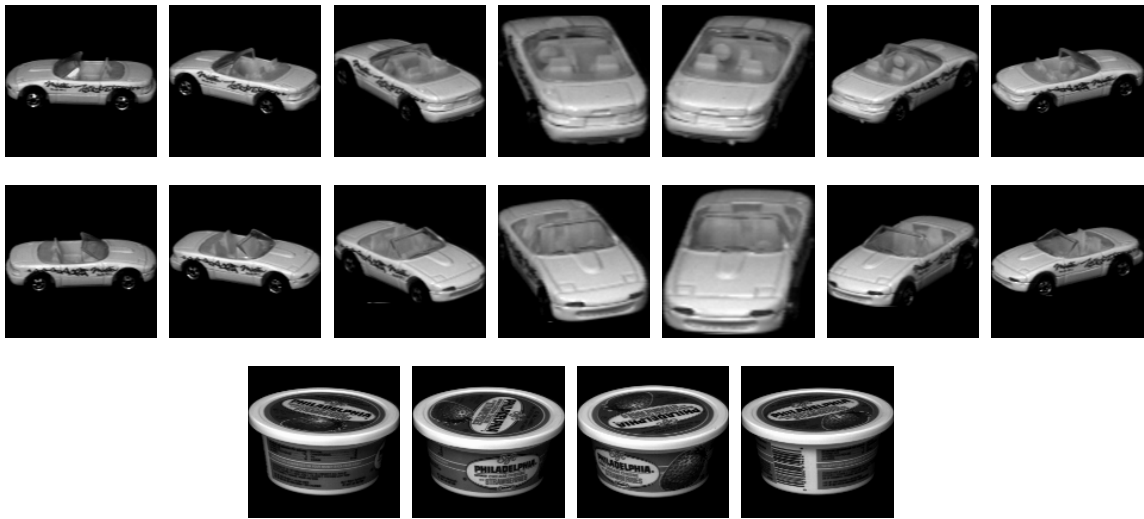


Figura 75: Protótipos usados em COIL

para gerar as associações entre vértices de G_i e G_m com o intuito de agilizar o processo de associação dado o tamanho elevado dessa base de testes.

5.2.4.1 Parâmetros de execução

Os parâmetros aplicados no algoritmo para os testes com essa base de imagens foram:

- $\lambda = 0,4$: aplicado na Equação 4.9
 - Implica em uma combinação do SCP (Eq. 4.6) e SCA_V (Eq. 4.8) aplicada na contribuição de vértices.
- $\sigma = 0$: usado na Equação 4.19
 - Indica que o algoritmo utiliza como critério de avaliação global a função $c_{eGlobal_{02}}$ definida pela Equação 4.18.
- $N = 10$: aplicado na comparação em níveis definida na Seção 4.3.2
 - Dessa forma foram usados 10 níveis de comparação no cálculo da contribuição de arestas na Equação 4.22.
- $\alpha = 0,4$: aplicado na Equação 4.1
 - Esse valor indica que a contribuição de arestas é ligeiramente maior que a contribuição de vértices na associação de pontos entre G_i e G_m .

Utilizamos a distância mínima de associação igual a 10 configurada na geração das associações entre vértices de acordo com o código do Algoritmo 7.

5.2.4.2 Metodologia de testes

Em seu trabalho, [3] procedeu com a classificação de todos os dígitos de teste utilizando todos os dígitos de treinamento. Em seus testes, [3] verifica o comportamento de seu algoritmo pela variação de k no seu classificador K-NN proposto. Contudo, nós avaliaremos nosso algoritmo pela variação no número de protótipos (como na seção anterior).

Nós faremos uma comparação entre os resultados obtidos em nossa execução usando o número máximo de protótipos para essa base (60000) com os resultados obtidos por [3] onde o autor também executa testes usando o número máximo de protótipos. Esses resultados encontram-se disponíveis em seu trabalho.

Para os testes efetuados com a base de imagens MNIST [94] nós selecionamos as primeiras 1000 imagens de teste para serem classificadas de acordo com os dígitos nas imagens de treinamento.

5.2.4.3 Resultados computacionais

Após a execução, com a variação no número de protótipos, nosso algoritmo encontrou os resultados dispostos no gráfico da Figura 76 onde pode-se perceber que a taxa de erro decresce com o aumento do número de protótipos.

Para a execução com o número máximo de dígitos de treinamento, [3] obteve 23 erros para os mesmos 1000 dígitos de teste enquanto nosso algoritmo encontrou 47 erros. O melhor desempenho obtido pelo algoritmo implementado em [3] enfatiza o uso a abordagem estatística onde a qualidade das associações é distribuída pelo grafo através da troca de mensagens entre os vértices.

As classificações erradas de nosso algoritmo podem ser visualizadas na Tabela 6 onde estão dispostas as imagens dos dígitos e acima de cada uma seu rótulo (a esquerda) e a classificação (a direita) dada por nosso algoritmo.

5.2.4.4 Testes para a distância mínima de associação

Como foi dito, nessa bateria de testes utilizamos a distância mínima de associação no processo de associação entre vértices, esperando diminuir o tempo de execução do nosso

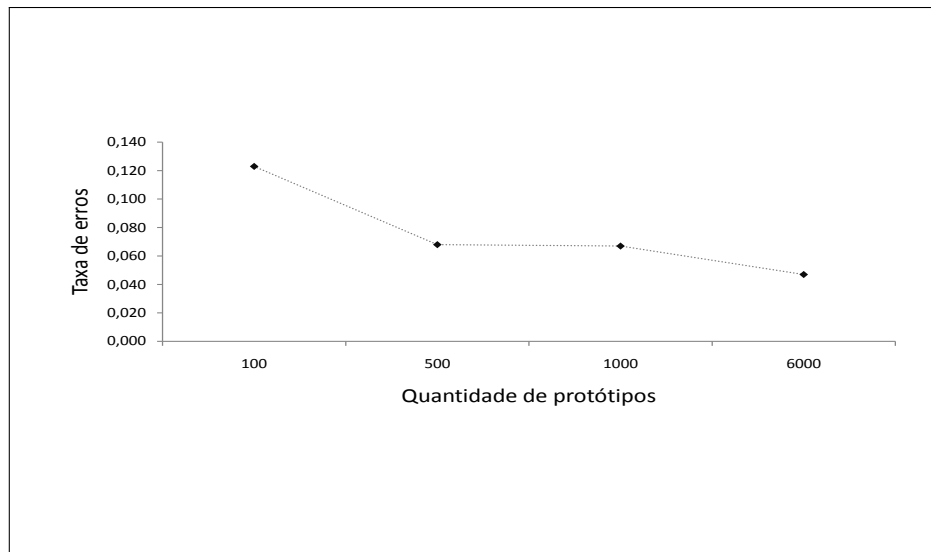


Figura 76: Taxa de erros em MNIST com variação no número de protótipos

algoritmo.

Para avaliar o ganho em relação ao tempo de execução propomos executar testes sobre 7 imagens de entrada, usando os primeiros 1000 protótipos da base de treinamento em que avaliamos os tempos de execução com a variação no tamanho da distância mínima de associação.

A Tabela 7 mostra os tempos (em segundos) para as execuções propostas com a variação na distância mínima de associação e o gráfico ilustrado na Figura 77 ilustra o comportamento do algoritmo onde nota-se um acréscimo de tempo de aproximadamente 270% em relação à execução com a menor d_{ma} .

Os resultados dispostos na Tabela 7 indicam a variação em relação ao tempo de execução do algoritmo quando submetido à diferentes valores para a distância mínima de associação. Notamos que esses tempos sofrem um acréscimo significativo quando a distância varia entre 5 e 20 pixels se estabilizando depois. Isso nos leva a crer que o uso dessa restrição por distância deve ser suficientemente pequeno para que possa reduzir o tempo de execução e suficientemente grande para que um vértice seja capaz de gerar associações.

Tabela 6: Erros encontrados na execução para MNIST. Temos a imagem e acima dela seu rótulo (a esquerda) e a nossa classificação (a direita).














































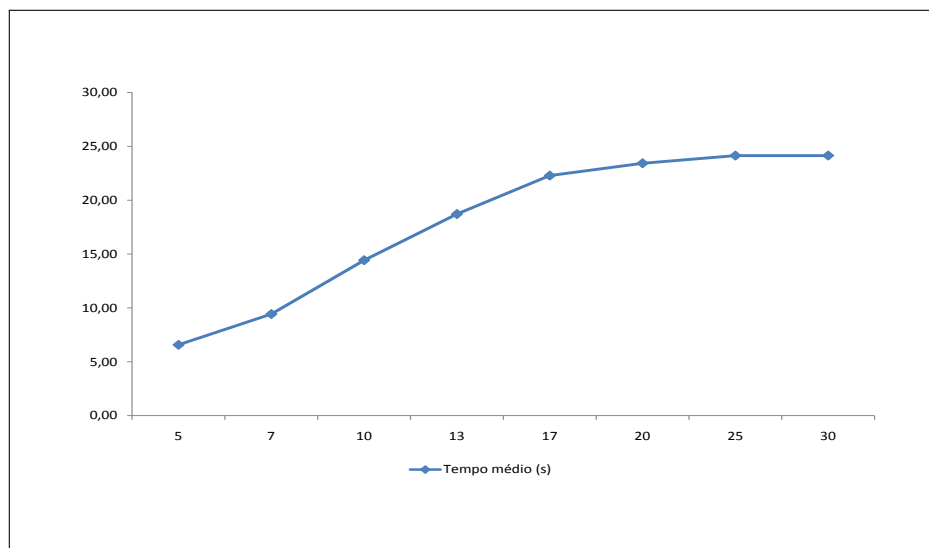
18:3→0	149:2→9	151:9→5	159:4→9	187:5→3	200:3→0	241:9→8
						
247:4→6	250:4→9	259:6→0	266:8→0	320:9→7	321:2→7	340:5→6
						
400:2→0	406:5→0	445:6→0	447:4→9	449:3→5	468:7→3	471:9→7
						
492:2→3	495:8→0	497:4→8	511:4→7	523:1→8	531:3→2	582:8→3
						
629:2→0	674:5→3	691:8→9	707:4→9	712:4→9	716:1→7	740:4→9
						
748:4→9	827:4→9	833:9→7	839:8→0	868:2→0	894:3→8	938:3→5
						
939:2→0	947:8→9	956:1→6	965:6→0	982:3→2		
						

Tabela 7: Tempos de execução com variação da distância mínima de associação

Dígito	Distância mínima de associação (em pixels)							
	5	7	10	13	17	20	25	30
	Tempos de execução em segundos							
1	7	10	16	21	26	28	28	28
2	6	10	15	20	23	23	24	24
3	7	11	16	22	26	28	29	29
4	5	7	11	13	15	16	16	16
5	7	9	16	20	24	25	26	25
6	7	9	13	17	21	22	23	24
7	7	10	14	18	21	22	23	23
Tempo médio (s)	6,57	9,43	14,43	18,71	22,29	23,43	24,14	24,14

Figura 77: Variação nos tempos de execução com diferentes d_{ma} 's

6 *Conclusões e Trabalhos Futuros*

Este trabalho apresenta uma técnica de reconhecimento de padrões, fundamentada no paradigma estrutural, que foi aplicada ao problema de reconhecimento de formas em imagens usando o conceito de casamento de padrões de pontos. A estratégia apresentada transcreve as características de um objeto em uma imagem através da modelagem por grafos, os quais são as estruturas de tratamento usadas para essa abordagem.

Os vértices de um grafo são denotados como pontos e dois conjuntos de pontos, representando dois grafos, são associados vértice a vértice caracterizando uma associação entre os grafos e, por conseguinte, uma associação entre imagens.

Neste trabalho, com os algoritmos implementados, foi mostrado o uso de uma estratégia gulosa empregada para a geração de uma solução inicial a qual é submetida a um processo de pós-otimização usando uma heurística *2-opt*. Também foram empregadas duas adaptações para a técnica de *Shape Context*, a primeira, aqui chamada de *Shape Context Percentual*, visa alterar a imagem da função original, mantendo o resultado dentro do intervalo $[0 \dots 1] \subset \mathbb{R}$, e a segunda, chamada de *Shape Context de Aresta*, capta os dados relativos aos pontos que preenchem os *bins* característicos através do ponto médio de cada aresta no grafo e estabelece o sentido do eixo de crescimento desses *bins* numa direção perpendicular a aresta selecionada naquele momento.

Além dessas adaptações utilizadas para a contribuição de vértices na função objetivo adotada neste trabalho, foi elaborado um procedimento para a contribuição de arestas baseado no uso de uma avaliação multicritério que denota a medida de dissimilaridade entre arestas. A principal vantagem desse método é a possibilidade de se modificar as funções que avaliam os critérios de similaridade entre duas arestas para se adequarem da melhor maneira para cada aplicação.

Também foi proposto um método para a avaliação de arestas que agrega valores não somente das arestas diretamente ligadas a um vértice em que se deseja comparar mas sim, do caminho gerado pelas subárvores do grafo que possuem como raiz o vértice escolhido no

momento da comparação sendo a profundidade dessa árvore definida como a quantidade de níveis a serem verificados.

Avaliou-se neste trabalho a eficiência com a qualidade dos resultados obtidos pelos algoritmos implementados através da comparação com técnicas presentes na literatura. Dessa forma, foi possível notar que o primeiro algoritmo, aplicado à base de dados KIMIA, possui um bom desempenho quanto às melhores classificações tendo encontrado 100% de acertos para a primeira classificação em todos os testes. Não obstante, esse desempenho cai muito rapidamente com o afastamento das primeiras classificações, o que nos leva a concluir que nosso algoritmo, ao ser aplicado à classificadores K-NN com um K elevado, torna-se ineficiente.

Nós também desenvolvemos uma técnica voltada à comparação de arestas em grafos de linha gerados a partir dos grafos originais. Com essa proposta nós amenizamos problemas encontrados com relação a diferenças geométricas obtidas por rotações em imagens de uma mesma classe e, ao aplicamos testes sobre imagens de silhuetas, averiguamos uma melhora na qualidade das classificações quando comparada a proposta inicial de nosso trabalho além de obter resultados semelhantes aos melhores resultados que encontramos na literatura para os mesmos testes efetuados.

A aplicação do nosso algoritmo para a base de imagens MNIST, onde selecionamos apenas uma parte do conjunto total de imagens para testes, apresentou o pior desempenho dentro dos testes computacionais efetuados. Comparamos os nossos resultados com os obtidos em [3] e ao aplicarmos nosso algoritmo usando o conjunto total de treinamento obtemos um resultado duas vezes pior que o encontrado por [3]. Nós avaliamos esse resultado como sendo um reflexo da etapa de aquisição de informações das imagens. Uma vez que nosso algoritmo é fundamentado no reconhecimento estrutural, a contribuição da etapa de pré-processamento é essencial para construirmos grafos com boa qualidade de informações. As imagens dessa base são muito pequenas e a aquisição de informações das bordas deve ser melhorada possibilitando a construção de melhores grafos.

Os melhores resultados foram obtidos para a base de imagens COIL onde melhoramos a taxa de acertos do trabalho de [3] de 0.0581 para 0,0338 usando 8 visões igualmente espaçadas para cada objeto e reduzindo de 0,0161 para 0,0040 quando usamos o conceito de *k-medoids*. Ressaltamos que para essa base usamos um classificador 1-NN assim como em [3] além de termos usados os mesmos protótipos do seu trabalho.

Através dos testes efetuados podemos concluir que ao trabalharmos com a contribuição de vértices custeada pela função *SCA* usando o mapeamento através de grafos de linha,

quando as imagens pertencentes a uma mesma classe sofrem variações geométricas, principalmente em relação à rotação, pode-se amenizar erros de associação causados por essas discrepâncias.

O uso da distância mínima de associação proposta como técnica para melhorar o tempo de execução do algoritmo mostrou ser eficaz, de acordo com os testes empregados na base de imagens MNIST.

Concluimos com este trabalho que, para imagens que possuam classes bem definidas e possam ter características bem extraídas para formarem os grafos correspondentes, proporcionam melhores classificações do que imagens de uma mesma classe que possuam características que destoam umas das outras, características essas como rotação 2D, por exemplo.

Propomos como trabalhos futuros, estabelecer novos testes sobre a base de imagens MNIST usando uma melhor forma de captação de dados das imagens, como por exemplo a ampliação das mesmas antes da tarefa de aquisição das informações de bordas, uma vez que são muito pequenas e não possuem características de bordas tão bem definidas quanto as imagens de KIMIA.

Propomos também, abordar mais minuciosamente os testes aplicados à base de imagens MNIST, averiguando o uso apenas do *SCA* para a contribuição de vértices dos grafos de linha.

Além dessas propostas é interessante proceder com testes sobre os valores dos parâmetros empregados no algoritmo implementado pois, como o número de parâmetros é relativamente elevado, podem ser observadas melhorias apenas ao mudarmos alguns dos valores usados neste trabalho.

E finalmente, investigar a contribuição de invariantes espectrais nas técnicas para reconhecimento de formas aqui estudadas.

Referências

- [1] CAMPOS, T. E. de. *Técnicas de seleção de características com aplicações em reconhecimento de faces*. Dissertação (Mestrado em Ciência da Computação) — Universidade de São Paulo, São Paulo, 2001.
- [2] DUDA, R. O.; HART, P. E.; STORK, D. G. *Pattern Classification (2nd Edition)*. 2. ed. [S.l.]: Wiley-Interscience, 2001.
- [3] NOMA, A. *Duas abordagens para casamento de padrões de pontos usando relações espaciais e casamento entre grafos*. Tese (Doutorado em Ciência da Computação) — Universidade de São Paulo, São Paulo, 2010.
- [4] SEBASTIAN, T. B.; KLEIN, P. N.; KIMIA, B. B. Recognition of shapes by editing their shock graphs. *IEEE Trans. Pattern Anal. Mach. Intell.*, v. 26, p. 550–571, 2004.
- [5] WOLFSON, H. J.; RIGOUTSOS, I. Geometric hashing: An overview. *IEEE Comput. Sci. Eng.*, v. 4, n. 4, p. 10–21, oct. 1997.
- [6] SEBASTIAN, T. B.; KLEIN, P. N.; KIMIA, B. B. Recognition of shapes by editing shock graphs. In: *In IEEE International Conference on Computer Vision*. [S.l.: s.n.], 2001. p. 755–762.
- [7] TURK, M. A.; PENTLAND, A. P. Face recognition using eigenfaces. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Maui, jun. 1991.
- [8] SILVA, P. Q. da; ROSA, A. N. de C. S. Reconhecimento facial aplicado a perícia criminal. In: *CONFERENCIA INTERNACIONAL DE PERICIAS EM CRIMES CIBERNETICOS (ICCyber)*. Brasília, [s.n.], 2004.
- [9] LOWE, D. G. Object recognition from local scale-invariant features. *Proceedings of the International Conference on Computer Vision*, v. 2, p. 1150–1157, jun. 1999.
- [10] SILVA-JR, W. S. da. *Reconhecimento de padrões utilizando filtros de correlação com análise de componentes principais*. Tese (Doutorado em Engenharia Elétrica) — Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2010. Doutorado em Engenharia Elétrica.
- [11] FORSYTH, D. A.; PONCE, J. *Computer Vision: A Modern Approach*. 1. ed. [S.l.]: Prentice Hall, 2002.
- [12] ZEILHOFER, P. et al. Técnicas de reconhecimento de formas para identificação de áreas de irrigação por imagens de satélite. In: *SIMPÓSIO BRASILEIRO DE SENSORIAMENTO REMOTO*, 2007. Florianópolis, 2007. p. 6291–6298.
- [13] BRONSTEIN, M. M.; BRONSTEIN, A. M. Shape recognition with spectral distances. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 33, n. 5, p. 1065–1071, maio 2011.

- [14] FELZENSZWALB, P.; SCHWARTZ, J. Hierarchical matching of deformable shapes. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, p. 1–8, 2007.
- [15] BELONGIE, S.; MALIK, J.; PUZICHA, J. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 24, n. 24, p. 509–522, abr. 2002.
- [16] CAETANO, T. S. *Graphical Models and Point Set Matching*. Tese (Ph.D. thesis) — UFRGS, Porto Alegre, RS, BRA, 2004.
- [17] JESUS, L. G. C. S. de. *Método de descida para problemas de otimização multiobjetivo*. Dissertação (Mestrado em Matemática) — Universidade Federal de Goiás, Goiânia, 2010.
- [18] BRANDSTETTER, M. C. G.; FREITAS, A. A. F. O uso da metodologia multicritério de apoio à decisão na definição da compra de um imóvel - a fase de estruturação. In: *ENEGEP97 - XVII Encontro Nacional de Engenharia de Produção e III Congresso Internacional de Engenharia Industrial*. Gramado, RS: [s.n.], 1997.
- [19] GOLDBARG M. C.; LUNA, H. P. L. *Otimização combinatória e programação linear: modelos e algoritmos*. 2ed. ed. [S.l.]: Editora Campus, 2000.
- [20] RODRIGUES, M. A. P. *Problema do caixeiro viajante - um algoritmo para resolução de problemas de grande porte baseado em busca local dirigida*. Dissertação (Mestrado em Engenharia) — Universidade Federal de Santa Catarina, Florianópolis, SC, 2000.
- [21] WATANABE, S. *Pattern recognition: human and mechanical*. New York, NY, USA: John Wiley & Sons, Inc., 1985.
- [22] KPALMA, K.; RONSIN, J. An overview of advances of pattern recognition systems in computer vision. In: _____. I-Tech Education and Publ., 2007. (Vision Systems: Segmentation and Pattern Recognition), cap. 10, p. 169–194. Teste. Disponível em: <http://cdn.intechopen.com/pdfs/324/InTech-An_overview_of_advances_of_pattern_recognition_systems_in_computer_vision.pdf>.
- [23] THEODORIDIS, S.; KOUTROUMBAS, K. *Pattern Recognition, Second Edition*. [S.l.]: Academic Press, Inc., 2003.
- [24] JAIN, A. K.; DUIN, R. P. W.; MAO, J. Statistical pattern recognition: A review. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, v. 22, n. 1, p. 4–37, 2000.
- [25] KUNCHEVA, L. I. *Combining Pattern Classifiers: Methods and Algorithms*. [S.l.]: Wiley-Interscience, 2004.
- [26] LIU, J.; SUN, J.; WANG, S. Pattern recognition: An overview. *IJCSNS International Journal of Computer Science and Network Security*, v. 6, n. 6, p. 57–61, jun. 2006.
- [27] KEUSTERMANS, J. et al. Image segmentation using graph representations and local appearance and shape models. In: *Proceedings of the 7th IAPR-TC-15 International Workshop on Graph-Based Representations in Pattern Recognition*. Berlin, Heidelberg: Springer-Verlag, 2009. (GbRPR '09), p. 353–365.

- [28] ZANA, Y.; MENA-CHALCO, J. P.; CESAR-JR., R. M. A novel polar-based human face recognition computational model. *Brazilian Journal of Medical and Biological Research*, v. 42, n. 7, p. 637–646, 2009. Disponível em: <<http://www.vision.ime.usp.br/creativision/publications/pdf/BJMBR-PHFR.pdf>>.
- [29] WACHENFELD, S. et al. Graph-based registration of partial images of city maps using geometric hashing. In: *Proceedings of the 7th IAPR-TC-15 International Workshop on Graph-Based Representations in Pattern Recognition*. Berlin, Heidelberg: Springer-Verlag, 2009. (GbRPR '09), p. 92–101.
- [30] BENGIOETXEA, E. et al. Inexact Graph Matching by Means of Estimation of Distribution Algorithms. *Pattern Recognition*, v. 35, p. 2867–2880, 2002.
- [31] NEUMANN, J.; SCHNÖRR, C.; STEIDL, G. Svm-based feature selection by direct objective minimisation. In: *DAGM-Symposium*. [S.l.]: Springer, 2004. (Lecture Notes in Computer Science, v. 3175), p. 212–219.
- [32] ZHENG, L.; HE, X. Classification techniques in pattern recognition. In: *Proceedings of the 13th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*. [S.l.: s.n.], 2005. p. 77–79.
- [33] GRACIANO, A. B. V. *Rastreamento de objetos baseado em reconhecimento estrutural de padrões*. Dissertação (Mestrado em Ciências) — Universidade de São Paulo, São Paulo, SP, mar. 2007.
- [34] DING, L.; GOSHTASBY, A. On the canny edge detector. *Pattern Recognition*, v. 34, p. 721–725, 2001.
- [35] NEVES, S. R. *Algoritmos para segmentação de imagens infravermelhas*. Tese (Doutorado em Engenharia Elétrica) — Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, jun. 2003.
- [36] FILHO Ôge M.; NETO, H. V. *Processamento Digital de imagens*. 1. ed. [S.l.: s.n.], 1999.
- [37] BHATTACHARYYA, S.; MAULIK, U.; DUTTA, P. Multilevel image segmentation with adaptive image context based thresholding. *Appl. Soft Comput.*, v. 11, n. 1, p. 946–962, jan. 2011.
- [38] CONSULARO, L. A. et al. Oversegmentation control for inexact graph matching: first results. In: *Proc. International Symposium on Mathematical Morphology (ISMM-07)*. [S.l.: s.n.], 2007.
- [39] MICHAILOVICH, O. V.; RATHI, Y.; TANNENBAUM, A. Image segmentation using active contours driven by the bhattacharyya gradient flow. *IEEE Transactions on Image Processing*, v. 16, n. 11, p. 2787–2801, 2007.
- [40] AYED, I. B. et al. Graph cut segmentation with a global constraint: Recovering region distribution via a bound of the bhattacharyya measure. In: *CVPR*. [S.l.]: IEEE, 2010. p. 3288–3295.
- [41] FREEDMAN, D.; TUREK, M. W. Graph cuts with many-pixel interactions: Theory and applications to shape modelling. *Image Vision Comput.*, v. 28, n. 3, p. 467–473, 2010.

- [42] BOYKOV, Y. Y.; JOLLY, M.-P. Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. *Computer Vision, IEEE International Conference on*, v. 1, p. 105, 2001.
- [43] NOMA, A. et al. A new algorithm for interactive structural image segmentation. *CoRR*, abs/0805.1854, 2008.
- [44] PORIKLI, F. Automatic image segmentation by wave propagation. In: *Proceedings of IS&T/SPIE Symposium on Electronic Imaging*. [S.l.: s.n.], 2004.
- [45] PENG, B.; ZHANG, L.; ZHANG, D. Automatic Image Segmentation by Dynamic Region Merging. *Image Processing, IEEE Transactions on*, v. 20, p. 3592–3605, 2011.
- [46] KLAVA, B. *Segmentação interativa de imagens via transformação watershed*. Dissertação (Mestrado em Ciência da Computação) — Instituto de Matemática e Estatística - Universidade de São Paulo, São Paulo, SP, jul. 2009.
- [47] KUMAR K B RAJA, V. K. R. H. C. S.; PATNAIK, L. M. Automatic image segmentation using wavelets. *International Journal of Computer Science and Network Security*, v. 9, n. 2, p. 305–313, 2009.
- [48] SHIH, F. Y.; CHENG, S. Automatic seeded region growing for color image segmentation. *Image Vision Comput.*, v. 23, n. 20, p. 877–886, sep 2005.
- [49] NIELSEN, F.; NOCK, R. Fast graph segmentation based on statistical aggregation phenomena. In: IKEUCHI, K. (Ed.). *Proceedings of the 10th IAPR Conference on Machine Vision Applications*. Tokyo, Japan: IAPR MVA Conference Committee, 2007. p. 150–153.
- [50] ZHANG, L.; JI, Q. Image segmentation with a unified graphical model. *IEEE Trans. Pattern Anal. Mach. Intell.*, v. 32, n. 8, p. 1406–1425, 2010.
- [51] NOMA, A.; PARDO, A.; CESAR-JR, R. M. Structural matching of 2d electrophoresis gels using graph models. In: *Proceedings of the 2008 XXI Brazilian Symposium on Computer Graphics and Image Processing*. Washington, DC, USA: IEEE Computer Society, 2008. (SIB-GRAPI '08), p. 71–78.
- [52] ROGERS, M.; GRAHAM, J.; TONG, R. 2 dimensional electrophoresis gel registration using point matching and local image-based refinement. In: HOPPE, A.; BARMAN, S.; ELLIS, T. (Ed.). *Proceedings of the British Machine Vision Conference*. [S.l.]: BMVA Press, 2004. p. 59.1–59.10.
- [53] HASHIMOTO, M.; CESAR-JR, R. M. Object detection by keygraph classification. In: TORSELLO, A.; ESCOLANO, F.; BRUN, L. (Ed.). *Proceedings of the 7th IAPR-TC-15 International Workshop on Graph-Based Representations in Pattern Recognition*. [S.l.]: Springer, 2009. (Lecture Notes in Computer Science, v. 5534), p. 223–232.
- [54] BELONGIE, S.; MALIK, J.; PUZICHA, J. Matching shapes. In: *ICCV*. [S.l.: s.n.], 2001. p. 454–463.
- [55] BOERES, M. C. S. *Heurísticas para Reconhecimento de Cenas por Correspondência de Grafos*. Tese (Doutorado em Engenharia de Produção) — Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, 2002.

- [56] LIU, G. et al. Unsupervised object segmentation with a hybrid graph model (hgm). *IEEE Trans. Pattern Anal. Mach. Intell.*, v. 32, n. 5, p. 910–924, maio 2010.
- [57] CONSULARO, L. A.; CESAR-JR, R. M.; BLOCH, I. Structural image segmentation with interactive model generation. In: *ICIP (6)*. [S.l.: s.n.], 2007. p. 45–48.
- [58] KATONA, E. A graph based data model for graphics interpretation. In: *Proceedings of the 7th IAPR-TC-15 International Workshop on Graph-Based Representations in Pattern Recognition*. Berlin, Heidelberg: Springer-Verlag, 2009. (GbRPR '09), p. 72–81.
- [59] LATECKI, L. J.; LAKÄMPER, R.; ECKHARDT, U. Shape descriptors for non-rigid shapes with a single closed contour. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2000. p. 424–429.
- [60] RAFTOPOULOS, K. A.; KOLLIAS, S. D. The global-local transformation for noise resistant shape representation. *Comput. Vis. Image Underst.*, v. 115, n. 8, p. 1170–1186, 2011.
- [61] LING, H.; JACOBS, D. W. Using the inner-distance for classification of articulated shapes. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. San Diego, CA, USA: [s.n.], 2005. (CVPR'05, v. 2), p. 719–726.
- [62] NETTO, P. O. B. *Grafos: Teoria, Modelos, Algoritmos*. [S.l.]: Edgard Blucher, 2006.
- [63] BUNKE, H.; ALLERMANN, G. Inexact graph matching for structural pattern recognition. *Pattern Recognition Letters*, v. 1, p. 245–253, 1983.
- [64] OGNIIEWICZ, R.; KUBLER, O. Hierarchic voronoi skeletons. *Pattern Recognition*, v. 28, n. 3, p. 343–359, 1995.
- [65] BLUM, H. A transformation for extracting new descriptors of shape. *Models for the Perception of Speech and Visual Form*, p. 362–380, 1967.
- [66] OLIVEIRA, F. P. M.; TAVARES, J. M. R. S. Emparelhamento de contornos em imagens usando informação de curvatura e otimização baseada em programação dinâmica. *IEEE LATIN AMERICA TRANSACTIONS*, v. 7, n. 6, p. 703–712, 2009.
- [67] LING, H.; JACOBS, D. W. Shape classification using the inner-distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE Computer Society, Los Alamitos, CA, USA, v. 29, p. 286–299, 2007. ISSN 0162-8828.
- [68] WANG, L. From blob metrics to posture classification to activity profiling. *Pattern Recognition, International Conference on*, v. 4, p. 1051–4651, 2006.
- [69] FELZENSZWALB, P. F.; HUTTENLOCHER, D. P. Efficient belief propagation for early vision. *Int. J. Comput. Vision*, v. 70, p. 41–54, 2006.
- [70] NOMA, A.; VELHO, L.; CESAR JR, R. M. A computer-assisted colorization approach based on efficient belief propagation and graph matching. In: *Proceedings of the 14th Iberoamerican Conference on Pattern Recognition: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. [S.l.: s.n.], 2009. (CIARP '09).
- [71] CAETANO, T. S. et al. Graphical models and point pattern matching. *IEEE Trans. PAMI*, v. 28, n. 10, p. 1646–1663, out. 2006. Disponível em: <<http://www.lume.ufrgs.br/bitstream/handle/10183/27603/000585735.pdf?sequence=1>>.

- [72] ZHANG, Z. Iterative point matching for registration of free-form curves and surfaces. *Int. J. Comput. Vision*, v. 13, n. 2, p. 119–152, out. 1994.
- [73] CARDOZE, D. E.; SCHULMAN, L. J. Pattern matching for spatial point sets. In: *FOCS*. [S.l.]: IEEE Computer Society, 1998. p. 156–165.
- [74] CHUI, H.; RANGARAJAN, A. A new point matching algorithm for non-rigid registration. *Comput. Vis. Image Underst.*, v. 89, n. 2-3, p. 114–141, fev. 2003.
- [75] LEPETIT, V.; PILET, J.; FUA, P. Point matching as a classification problem for fast and robust object pose estimation. In: *Proceedings of the 2004 IEEE computer society conference on Computer vision and pattern recognition*. Washington, DC, USA: IEEE Computer Society, 2004. (CVPR'04), p. 244–250.
- [76] HANCOCK, E. R.; WILSON, R. C. Graph-based methods for vision: A yorkist manifesto. In: *Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*. [S.l.]: Springer-Verlag, 2002. p. 31–46.
- [77] LUO, B.; HANCOCK, E. R. Structural graph matching using the em algorithm and singular value decomposition. *IEEE Trans. Pattern Anal. Mach. Intell.*, v. 23, n. 10, p. 1120–1136, out. 2001.
- [78] CARCASSONI, M.; HANCOCK, E. R. Spectral correspondence for point pattern matching. *Pattern Recognition*, v. 36, n. 1, p. 193–204, 2003.
- [79] BURKARD E. ÇELA, P. P. R.; PITSOULIS, L. The quadratic assignment problem. In: _____. *Handbook of Combinatorial Problems*. Boston: Kluwer Academic, 1998. p. 241–338.
- [80] COMMANDER, C. W. A survey of the quadratic assignment problem, with applications. *Morehead Electronic Journal of Applicable Mathematic*, n. 4, p. 1–15, 2005. Disponível em: <<http://www2.moreheadstate.edu/files/colleges/science/mcs/mejam/Commander.pdf>>. Acesso em: 1 mar. 2012.
- [81] MARTINS, V. C. et al. Otimização de layouts industriais com base em busca tabu. In: _____. *Gestão e Produção*, 2003. v. 10, n. 1, p. 69–88. Disponível em: <<http://www.scielo.br/pdf/gp/v10n1/a06v10n1.pdf>>. Acesso em: 1 jun. 2012.
- [82] GAREY, M. R.; JOHNSON, D. S. *Computers and intractability: a guide to the theory of NP-completeness*. New York: W. H. Freeman and Company, 1979.
- [83] CESAR-JR, R. M. et al. Inexact graph matching for model-based recognition: Evaluation and comparison of optimization algorithms. *Pattern Recognition*, v. 38, p. 2099–2113, 2005.
- [84] BELONGIE, S.; MALIK, J.; PUZICHA, J. Shape context: A new descriptor for shape matching and object recognition. In: *In NIPS*. [S.l.: s.n.], 2000. p. 831–837.
- [85] MORI, G. et al. Efficient shape matching using shape contexts. *IEEE Trans. Pattern Analysis and Machine Intelligence*, v. 27, p. 1832–1837, 2005.
- [86] FORSTER, C. H. Q. *Alinhamento Imagem-Modelo Baseada na Visão Estéreo de Regiões Planares Arbitrárias*. Tese (Doutorado em Engenharia Elétrica) — FEEC-UNICAMP, Campinas, SP, 2004.

- [87] GERSTING, J. L. *Fundamentos Matemáticos Para a Ciência da Computação*. [S.l.]: LTC Editora, 1995.
- [88] KONAK, A.; COIT, D. W.; SMITH, A. E. Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering & System Safety*, v. 91, n. 9, p. 992–1007, sep 2006.
- [89] MARLER, R. T.; ARORA, J. S. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, v. 26, n. 6, p. 369–395, apr 2004.
- [90] PRESTES, A. N. *Uma análise experimental de abordagens heurísticas aplicadas ao problema do caixeiro viajante*. Dissertação (Mestrado em Sistemas e Computação) — Universidade Federal do Rio Grande do Norte, Natal, RN, jul. 2006.
- [91] DIESTEL, R. *Graph Theory (Graduate Texts in Mathematics)*. [S.l.]: Springer, 2005.
- [92] MURASE, H.; NAYAR, S. Learning and Recognition of 3D Objects from Appearance. In: *IEEE Workshop on Qualitative Vision*. [s.n.], 1993. p. 39–50. Disponível em: <http://www1.cs.columbia.edu/CAVE/publications/pdfs/Murase_QV93.pdf>.
- [93] MURASE, H.; NAYAR, S. K. Visual learning and recognition of 3-d objects from appearance. *Int. J. Comput. Vision*, v. 14, n. 1, p. 5–24, jan. 1995.
- [94] LECUN, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, v. 86, n. 11, p. 2278–2324, nov. 1998.
- [95] MOURATIDIS, K.; PAPADIAS, D.; PAPADIMITRIOU, S. Tree-based partition querying: a methodology for computing medoids in large spatial datasets. *The VLDB Journal*, v. 17, n. 4, p. 923–945, 2008.
- [96] MELLO, C. E. R. de. *Agrupamento de regiões: uma abordagem utilizando acessibilidade*. Dissertação (Mestrado) — UFRJ, 2008.