Modelagem e reconhecimento de objetos estruturados: uma abordagem estatístico-estrutural

Ana Beatriz Vicentim Graciano

Tese apresentada
ao
Instituto de Matemática e Estatística
da
Universidade de São Paulo
para
obtenção do título
de
Doutor em Ciências

Programa: Ciência da Computação
Orientador: Prof. Dr. Roberto Marcondes Cesar Jr.

Durante o desenvolvimento deste trabalho, a aluna recebeu auxílio financeiro da CAPES e do CNPq.

São Paulo, março de 2012.

Modelagem e reconhecimento de objetos estruturados: uma abordagem estatístico-estrutural

Esta tese trata-se da versão original apresentada pela aluna Ana Beatriz Vicentim Graciano, por ocasião do depósito de sua tese de doutorado em Ciência da Computação.

Ana Beatriz Vicentim Graciano: *Modelagem e reconhecimento de objetos estruturados: uma abordagem estatístico-estrutural*
Tese de Doutorado, © março de 2012

RESUMO

Título: Modelagem e reconhecimento de objetos estruturados: uma abordagem estatístico-estrutural

Esta tese de doutorado aborda os tópicos de modelagem e de reconhecimento de objetos estruturados, ou sistemas estruturados de objetos, em imagens. Um objeto ou sistema estruturado é aquele que pode ser descrito através de elementos primitivos que o compõem e pelas relações existentes entre esses elementos. Por exemplo, uma aeronave pode ser descrita pelos seguintes elementos primitivos: asas direita e esquerda, fuselagem e *cockpit*.

O aspecto relacional de um objeto estruturado direciona sua representação computacional e seu reconhecimento em imagens ao paradigma estrutural de reconhecimento de padrões. Contudo, a variabilidade das características dos seus elementos primitivos é melhor representada através do paradigma estatístico de reconhecimento de padrões. Devido à complementaridade dos paradigmas, a conjunção dessas abordagens é um tema de pesquisa de interesse atual.

Para conjugar esses dois aspectos, esta tese propôs uma metodologia que combina o conhecimento *a priori* das relações que caracterizam um objeto estruturado com dados estatísticos coletados de amostras desse objeto, num modelo híbrido denominado *grafo estatístico-relacional* (GER).

Segundo essa representação, foi estudada uma abordagem probabilística para reconhecer um objeto estruturado em imagens. Nesse cenário, o GER modelo é considerado uma variável aleatória, enquanto uma rotulação de uma imagem de entrada é interpretada como uma potencial observação do modelo. A tarefa de reconhecimento foi então formulada como um problema de otimização, que busca maximizar a probabilidade da observação de acordo com o modelo.

O método foi aplicado à modelagem de órgãos abdominais em imagens de ressonância magnética não-contrastadas. Esses órgãos apresentam um arranjo espacial consistente em imagens distintas, além de propriedades de aparência e anatômicas variáveis, o que vem ao encontro da proposta da representação por GER e da abordagem probabilística para o reconhecimento dos órgãos em novas imagens.

Palavras-chave: reconhecimento estrutural de objetos, reconhecimento estatístico de objetos, modelagem de objetos estruturados, grafo relacional com atributos, função objetivo, segmentação

ABSTRACT

Title: Modeling and recognition of structured objects: a statistical-relational approach

The purpose of this thesis was to propose a formalism for the problems of modeling and recognition of a structured object, or a system of structured objects, in images. A structured object is one that may be described in terms of its compound primitive elements and their inherent relations. For instance, an aircraft may be described in terms of the following primitives: right and left wings, fuselage, and cockpit.

The relational aspect of structured objects leads these problems to solutions in structural pattern recognition, which describes patterns as primitives and relations. Nevertheless, the variability of primitive elements and of their relations is better modeled by traditional statistical pattern recognition methods. Because of the complementary capabilities of these approaches, the fusion of both has recently been pointed out as a trend in computer vision.

To consider these sources of information, the methodology presented herein combines relational cues inherent to a structured object with statistical information learned from a set of object samples. A hybrid model of a structured object is represented by means of a *statistical relational graph* (SRG). The SRG is a prototype attributed relational graph (ARG) in which nodes represent primitive elements and arcs link nodes representing related primitives. Each node or arc is associated with attributes which are parameters of probability distributions that describe random variables representing primitive or relational attributes.

Based on this representation, a probabilistic approach was proposed to tackle the problem of recognizing a structured object in an input image. The model SRG is interpreted as a random variable, whereas a labeling of the input image is considered a potential observation of the model. The recognition task was formulated as the optimization of an objective-function that is actually a probability measure to be maximized.

The proposed approach was applied to the modeling of abdominal organs in non-contrasted magnetic resonance images. These organs present consistent spatial arrangement in distinct images, as well as varying appearance and anatomical properties, which meet the principle of the SRG representation and the associated probabilistic recognition scenario.

Keywords: structural pattern recognition, statistical pattern recognition, structured object modeling, attributed relational graph, objective function, segmentation

SUMÁRIO

Lista de abreviaturas xi
Lista de símbolos xii
Lista de Figuras xiii
Lista de Tabelas xv
1 introdução 1
1.1 Considerações sobre reconhecimento de objetos 1
1.1.1 Abordagens estatística e estrutural para reconhecimento de objetos 2
1.2 Abordagem híbrida para reconhecimento de objetos estruturados 3
1.2.1 Objetivos 3
1.2.2 Contribuições 5
1.3 Organização do Texto 7
2 revisão bibliográfica 9
2.1 Definições preliminares 9
2.1.1 Conceitos elementares de teoria dos grafos . . . 9
2.1.2 Grafos e representação de dados pictóricos . . . 10
2.2 Abordagens híbridas para reconhecimento de objetos . 12
2.2.1 Métodos híbridos e geração de modelo 13
2.2.2 Métodos híbridos e casamento entre grafos . . . 15
2.3 Tendências no estudo de casamento entre grafos 16
3 modelagem e reconhecimento de objetos estruturados 19
3.1 Definições e notações 20
3.2 Modelagem de objetos estruturados 21
3.2.1 Modelo conceitual 21
3.2.2 Modelo computacional: grafo estatístico-relacional 22
3.3 Reconhecimento estatístico-estrutural baseado em modelo 25
3.3.1 Da representação digital da imagem ao reconhecimento segundo modelo 27
3.3.2 Grafo de observação 28
3.3.3 Formulação probabilística 30
3.3.4 Hipótese de independência mútua 33
3.3.5 Hipótese de independência condicional 34
3.4 Resultados experimentais 37
3.4.1 Dados simulados 37
3.4.2 Configurações do experimento 39
3.4.3 Simulação 1 42
3.4.4 Simulação 2 47
3.4.5 Simulação 3 52
3.4.6 Análise das simulações 57

4 APLICAÇÕES 59
4.1 Modelagem e segmentação de imagens de ressonância magnética . . . 59
4.2 Dados de ressonância magnética abdominal . . . 62
4.3 O modelo abdominal . . . 63
4.3.1 Definições e notações . . . 63
4.3.2 Descrição conceitual . . . 64
4.3.3 Propriedades dos órgãos abdominais e suas relações . . . 64
4.4 Segmentação de órgãos abdominais em IRM . . . 67
4.4.1 Representação de uma segmentação por grafo de observação . . . 68
4.4.2 Representação computacional do modelo conceitual . . . 70
4.4.3 Segmentação probabilístico-estrutural dos órgãos abdominais . . . 76
4.5 Resultados experimentais . . . 80
4.5.1 Grafo estatístico-relacional . . . 81
4.5.2 Segmentações ideais . . . 83
4.5.3 Variações sobre as segmentações ideais . . . 87
4.5.4 Avaliação das segmentações . . . 88
5 CONSIDERAÇÕES FINAIS 95
5.1 Resumo das contribuições . . . 95
5.1.1 Modelagem híbrida de objetos estruturados . . . 96
5.1.2 Abordagem probabilística para reconhecimento de objetos estruturados . . . 97
5.1.3 Ferramentas computacionais e implementação . . . 98
5.2 Perspectivas . . . 99
5.2.1 Desafios em relação ao modelo de objetos estruturados . . . 99
5.2.2 Desafios em relação ao processo de reconhecimento . . . 100
5.2.3 Desafios em relação à modelagem de órgãos computacionais . . . 100

ANEXOS 103
A LISTA DE PUBLICAÇÕES 105
Artigo SIBGRAPI 2008 106
Relatório Técnico 2009 114
Artigo Pattern Recognition 2011 141
Artigo Signal, Image and Video Processing 2012 162
ÍNDICE REMISSIVO 185
REFERÊNCIAS BIBLIOGRÁFICAS 187

LISTA DE ABREVIATURAS

ARG	*attributed relational graph*
FDG	grafo descrito por funções (*function-described graphs*)
GARG	grafo relacional genérico com atributos (*generalized attributed relational graph*)
GER	grafo estatístico-relacional
GRA	grafo relacional com atributos
GRG	grafo aleatório genérico (*generalized random graph*)
IRM	imagem de ressonância magnética
MAP	estimador de máximo a posteriori
MRF	campos aleatórios de Markov (*Markov random fields*)
RM	ressonância magnética
SRG	*statistical relational graph*

LISTA DE SÍMBOLOS

G_m	Grafo modelo
g_{obs}	Grafo de observação
V_m	Conjunto de nós de um grafo modelo
E_m	Conjunto de arcos de um grafo modelo
V_{obs}	Conjunto de nós de um grafo de observação
E_{obs}	Conjunto de arcos de um grafo de observação
$\|V_m\|, \|V_{obs}\|$	Número de nós no conjunto V_m/V_{obs}
$\|E_m\|, \|E_{obs}\|$	Número de arcos no conjunto E_m/E_{obs}
X_p	$p-$ésima variável aleatória representando um atributo de aparência
x_p	uma observação da variável aleatória X_p
Y_q	$q-$ésima variável aleatória representando um atributo relacional
y_q	uma observação da variável aleatória Y_q
$\overrightarrow{X}$	Vetor aleatório associado a um vértice de V_m
$\overrightarrow{Y}$	Vetor aleatório associado a um arco de E_m
$\overrightarrow{x}$	Vetor aleatório associado a um vértice de V_{obs}
$\overrightarrow{y}$	Vetor aleatório associado a um arco de E_{obs}
P	imagem bi ou tridimensional
R	domínio da imagem P no espaço de coordenadas
L	conjunto de elementos primitivos
L^*	conjunto de elementos primitivos e fundo
I	função de intensidade de P
S	partição de R
s_l	conjunto de regiões de S rotuladas com identificador $l \in L^*$
f	rotulação de S

LISTA DE FIGURAS

Figura 1 Representação estatística de objetos. 4
Figura 2 Representação estrutural de objetos. 5
Figura 3 Representação de objetos por grafo estatístico-relacional (GER). 23
Figura 4 Criação de um GER. 25
Figura 5 Pseudo-código para a criação de um GER (G_m). 26
Figura 6 Representação digital de uma imagem e modelo conceitual. 28
Figura 7 GER de observação. 29
Figura 8 Criação de um GRA de observação (g_{obs}). . . . 31
Figura 9 Casamento probabilístico de grafos. 32
Figura 10 Representação gráfica da simulação. 39
Figura 11 Simulação 1: densidades de probabilidade das variáveis aleatórias representantes das características de aparência de cada elemento primitivo modelado: l_1 (amarelo), l_2 (azul), l_3 (vermelho), l_4 (verde). 43
Figura 12 Resultados para Simulação 1: dados amostrados e respectivas curvas probabilidade $\times$ erros para $k = 1$ e $n = 1$. 44
Figura 13 Resultados para Simulação 1: dados amostrados e respectivas curvas probabilidade $\times$ erros para $k = 5$ e $n = 50$. 45
Figura 14 Resultados para Simulação 1: dados amostrados e respectivas curvas probabilidade $\times$ erros para $k = 100$ e $n = 25$. 46
Figura 15 Simulação 2: densidades de probabilidade das variáveis aleatórias representantes das características de aparência de cada elemento primitivo modelado: l_1 (amarelo), l_2 (azul), l_3 (vermelho), l_4 (verde). 48
Figura 16 Resultados para Simulação 2: dados amostrados e respectivas curvas probabilidade $\times$ erros para $k = 1$ e $n = 1$. 49
Figura 17 Resultados para Simulação 2: dados amostrados e respectivas curvas probabilidade $\times$ erros para $k = 5$ e $n = 50$. 50
Figura 18 Resultados para Simulação 2: dados amostrados e respectivas curvas probabilidade $\times$ erros para $k = 100$ and $n = 25$. 51

Figura 19 Simulação 3: densidades de probabilidade das variáveis aleatórias representantes das características de aparência de cada elemento primitivo modelado: l_1 (amarelo), l_2 (azul), l_3 (vermelho), l_4 (verde). 53
Figura 20 Resultados para Simulação 3: dados amostrados e respectivas curvas probabilidade $\times$ erros para $k = 1$ e $n = 1$. 54
Figura 21 Resultados para Simulação 3: dados amostrados e respectivas curvas probabilidade $\times$ erros para $k = 5$ e $n = 50$. 55
Figura 22 Resultados para Simulação 3: dados amostrados e respectivas curvas probabilidade $\times$ erros para $k = 100$ and $n = 25$. 56
Figura 23 Imagens de ressonância magnética contendo o abdome. 63
Figura 24 Rede semântica representando o abdome. . . . 65
Figura 25 Pseudo-código para calcular estatísticas consecutivas. 77
Figura 26 Imagens do conjunto de treinamento. 81
Figura 27 Imagens do conjunto de treinamento segmentadas. 82
Figura 28 Imagens distintas com regiões dilatadas correspondentes ao fígado. 87
Figura 29 Imagens distintas com rótulos trocados entre os rins direito e esquerdo. 88
Figura 30 Imagens distintas com erros aleatórios entre 1% e 10% adicionados. 89
Figura 31 Imagens distintas com erros aleatórios entre 10% e 100% adicionados. 90
Figura 32 Probabilidades das imagens segmentadas (resultado ideal, resultado com fígado dilatado, e resultado com rins trocados). 91
Figura 33 Probabilidades das imagens segmentadas com erros aleatórios entre 1% e 10%. 92
Figura 34 Probabilidades das imagens segmentadas com erros aleatórios entre 10% e 100%. 92

LISTA DE TABELAS

Tabela 1 Resumo comparativo entre os métodos de reconhecimento de padrões estatístico e estrutural. 4
Tabela 2 Órgãos e seus identificadores. 82
Tabela 3 Estatísticas dos atributos de aparência armazenadas em G_m. 83
Tabela 4 Estatísticas dos atributos relacionais armazenadas em G_m. 84
Tabela 5 Observações dos atributos de aparência das imagens de treinamento. 85
Tabela 6 Observações dos atributos relacionais das imagens de treinamento. 86

1 INTRODUÇÃO

classificação*:*
(...) 2: arranjo sistemático de elementos em grupos ou categorias de acordo com critérios pré-estabelecidos.
reconhecimento*:*
1: (...) b: conhecimento ou percepção de alguém ou algo presente e já visto antes. (...)

— Dicionário virtual *Merriam-Webster*

1.1 CONSIDERAÇÕES SOBRE RECONHECIMENTO DE OBJETOS

Segundo a língua portuguesa, *padrão* é "o que serve de base ou norma para avaliação; objeto que serve de modelo à feitura de outros" [Aur95]. Assim, *reconhecer* ou *classificar*[1] um objeto envolve o conhecimento de características discriminantes que o descrevem e a capacidade de utilizar tal descrição para avaliar outros objetos e caracterizá-los como exemplares desse conceito ideal. Por exemplo, uma vez que o conceito de *cão* faça parte do universo de conhecimento de uma pessoa, ela poderá reconhecer uma grande variedade de exemplos do mesmo animal, variáveis em termos de cor, espécie, tamanho, entre outras propriedades. Porém, todos os exemplares são observações concretas do ideal de mamífero que representam.

O processo de reconhecimento de um objeto permeia as mais diversas atividades humanas, desde aquelas que envolvem a sobrevivência (por exemplo, a detecção de um animal ameaçador ou de expressão de genes relacionados a uma dada doença) até outras mais cotidianas como encontrar o carro no estacionamento.

Embora essa tarefa seja realizada com facilidade por grande parte dos seres humanos, ela ainda constitui um desafio em visão computacional, devido sobretudo às circunstâncias diversas que podem ser retratadas numa imagem. Exemplos de tais dificuldades são características próprias do processo de aquisição de um certo tipo de imagem, oclusões totais ou parciais com outros objetos da cena, variações de iluminação, além da própria variedade de características que um exemplar de um objeto ideal pode apresentar, tais como texturas, cores, alterações de posicionamento e de forma.

Apesar de desafiador, reconhecer objetos em imagens é uma etapa elementar para extração de informações a partir de dados pictóricos,

1 Embora os termos reconhecer e classificar não possuam exatamente o mesmo significado, eles serão aplicados de forma intercambiável ao longo do texto.

e o sucesso do resultado obtido frequentemente valida ou inutiliza a aplicação de sistemas de visão artificial a finalidades diversas, como o processamento de imagens médicas, a biometria, a vigilância automatizada, dentre outros.

1.1.1 *Abordagens estatística e estrutural para reconhecimento de objetos*

Em visão computacional, o problema de reconhecimento de objetos em imagens compreende tanto a *representação* ou *modelagem* do objeto no computador, quanto sua *classificação* ou seu *reconhecimento* numa dada imagem. De modo geral, esses dois aspectos estão interligados e definem uma abordagem a ser adotada na resolução do problema.

O paradigma de reconhecimento estatístico de padrões [DHS00, JDM00] tem sido estudado ao longo de décadas, e consiste num dos mais antigos para reconhecimento de objetos. Suas técnicas estão fundamentadas sobre métodos matemáticos e estatísticos, e, devido aos conhecimentos adquiridos e à sua popularidade, suas limitações e vantagens são razoavelmente conhecidas. Nessa abordagem, um objeto é descrito através de um conjunto de n características ou medidas individuais, representadas por um ponto num espaço $n-$dimensional, e o problema de reconhecimento consiste na classificação desse conjunto de características observadas para um objeto segundo as classes de objetos representadas nesse espaço.

Se quisermos reconhecer um objeto *estruturado*, ou um *sistema estruturado* de objetos, podemos descrevê-lo de acordo com os elementos primitivos que o compõem e as relações entre esses elementos, o que também denominamos reconhecimento de objetos descritos por partes.

Para tratar objetos estruturados, ou sistemas estruturados de objetos, um outro paradigma, denominado reconhecimento estrutural de padrões [Pav80, Sch92], é mais adequado. Essa outra frente surgiu em meados dos anos 60, e propôs a representação dos elementos primitivos de um objeto estruturado através de um *grafo*, que é a representação típica adotada para padrões segundo esse formalismo. Essa representação permite descrever não apenas os objetos de acordo com suas características individuais, mas também as relações que compartilham com demais objetos numa cena.

As técnicas para reconhecimento estrutural de objetos envolvem a adoção de algoritmos que podem ser bastante custosos de um modo geral, pois recaem no problema de casamento entre grafos. Por esse motivo, esse paradigma foi pouco explorado durante um longo período, principalmente devido à limitação de recursos computacionais necessários para validá-lo e aplicá-lo na época. Contudo, nas últimas duas décadas, os métodos baseados em reconhecimento estrutural de padrões têm suscitado novo interesse, graças à existência de maquinário mais adequado e, principalmente, pelo poder de representação

que os grafos detêm, e também por toda a teoria que sustenta essa estrutura.

Um breve resumo comparativo de aspectos distintos dessas duas abordagens relevantes é apresentado na Tabela 1, enquanto um exemplo ilustrativo é dado nas Figuras 1 e 2 [2].

Na Figura 1, temos dois exemplos de objetos representados no paradigma estatístico. O coala é descrito por medidas de duas características: simetria bilateral e nível de cinza. Já o veleiro é descrito através de medidas de duas características: altura do mastro e comprimento do barco. Em ambos esses casos, os objetos são representados por um ponto num espaço bidimensional de características.

Na Figura 2, temos dois exemplos de objetos estruturados representados no paradigma estrutural. Cada elemento primitivo é representado por um nó do grafo, e as relações entre esses elementos são representadas por arcos conectando os nós. Nos exemplos, o avião é decomposto em quatro elementos primitivos: *cockpit*, asas direita e esquerda, e fuselagem. Esses elementos estão relacionados entre si da seguinte maneira: o *cockpit* está em frente à fuselagem, que está à direita da asa esquerda, e à esquerda da asa direita, e assim por diante. Já o veleiro, nessa representação estrutural, é decomposto em três elementos primitivos, vela, mastro, e barco, que se relacionam em parte da seguinte maneira: a vela é sustentada pelo mastro, que está acima do barco.

1.2 ABORDAGEM HÍBRIDA PARA RECONHECIMENTO DE OBJETOS ESTRUTURADOS

1.2.1 *Objetivos*

A modelagem e reconhecimento de objetos estruturados através do paradigma estrutural já havia sido tema de estudo de uma série de trabalhos do grupo de visão computacional [Gra07, Nom10, Pai10], sendo que alguns deles [PGCJHJ08, NGCJ+12] foram trabalhos colaborativos da aluna. Nesses dois últimos trabalhos, as características de aparência e relacionais de um objeto eram associadas aos vetores de atributos de nós e arcos de um grafo relacional com atributos (GRA) modelo. Contudo, esses atributos eram calculados a partir dos dados de uma única imagem. Portanto, a reutilização do modelo para reconhecer um dado objeto em imagens distintas tornava-se restrita, sendo aplicável, por exemplo, a imagens muito similares do mesmo objeto, como em quadros distintos de sequências de imagens digitais relativamente estáveis. Desse fato, surgiu então a idéia de considerar a dimensão estatística dos atributos e integrá-la ao modelo, bem como ao reconhecimento dos objetos.

2 Imagens obtidas do banco de imagens da Universidade de Berkeley (www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/)

Tabela 1: Resumo comparativo entre os métodos de reconhecimento de padrões estatístico e estrutural.

	Paradigma Estatístico	Paradigma Estrutural
Representação de padrões	espaço/vetor de características	primitivas e relações, gramática, grafo, grafo relacional com atributos
Métodos de classificação	classificador bayesiano, vizinhos mais próximos, mínima distância ao protótipo, etc.	inferência de gramáticas, casamento entre grafos (exato, inexato)
Histórico da área	pesquisa intensa desde a década de 60	interesse inicial na década de 60, mas avanços significativos concentrados nos últimos 15 anos
Desafios e limitações	redução de dimensionalidade, determinação de características discriminantes, incapacidade de representação de padrões com propriedades estruturais/relacionais relevantes	complexidade de tempo e espaço dos métodos de classificação

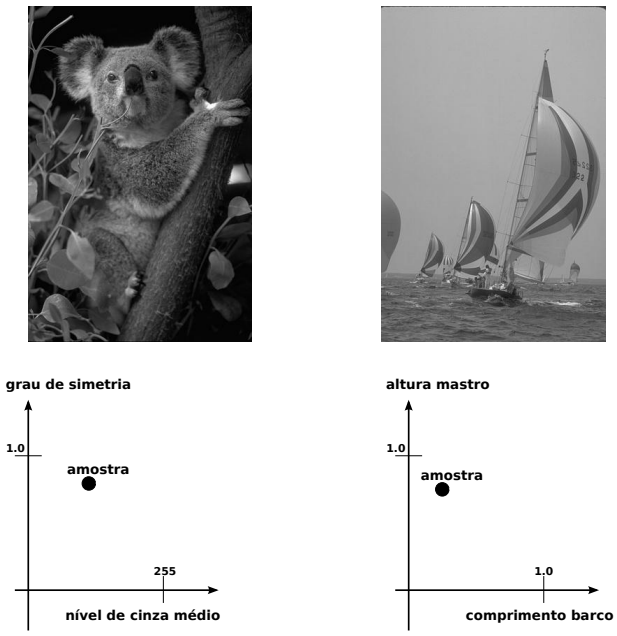


Figura 1: Representação estatística de objetos: veleiro e coala, e exemplos de características observadas para uma amostra de cada objeto, representadas no espaço de características.

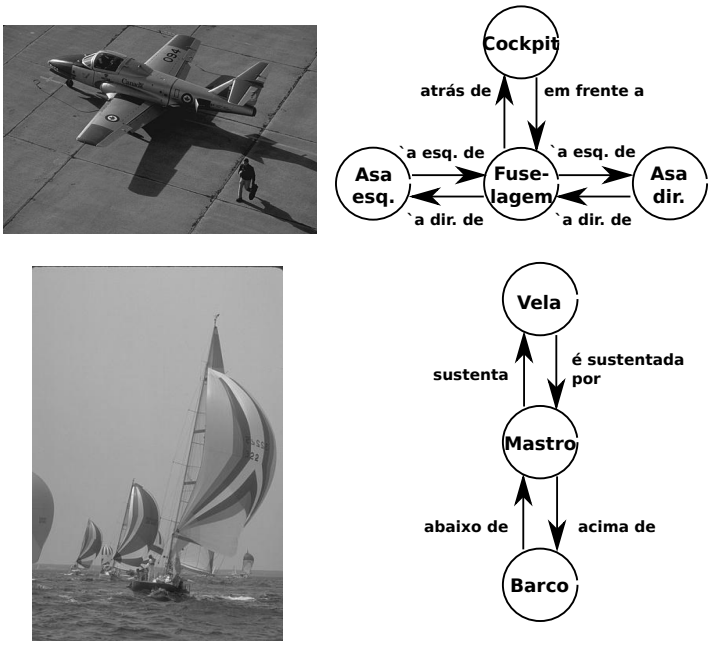


Figura 2: Representação estrutural de objetos: avião e veleiro, e exemplos de respectivos grafos relacionais. Cada nó representa um elemento primitivo que compõe o objeto, e cada arco indica uma propriedade relacional entre os elementos primitivos.

Essa idéia também foi corroborada pela tendência atual em buscar meios de integrar as abordagens estatística e estrutural de reconhecimento de padrões, de modo a conjugar os fundamentos e o conhecimento dos conceitos e algoritmos comumente usados em reconhecimento estatístico de padrões com a generalidade e abrangência da representação dos métodos estruturais baseados em grafos para reconhecimento de objetos em visão computacional [BIN05, CFH05, WH97, NB04, SAS03].

Portanto, nesta tese de doutorado, propusemos uma representação híbrida para objetos estruturados, através da qual descrevemos os elementos primitivos em termos de estatísticas de aparência e de relações, e desenvolvemos uma metodologia para reconhecimento desses objetos segundo um modelo, na qual ambas as fontes de conhecimento *a priori* são consideradas.

1.2.2 *Contribuições*

Para modelar uma classe de objetos estruturados, propusemos uma representação híbrida através de um grafo estatístico-relacional (GER). Essa representação consiste num grafo relacional com atributos (GRA), em que cada nó corresponde a um elemento primitivo, e cada arco conecta pares de nós que se relacionam de alguma forma. Assim, a

estrutura do grafo reflete as relações espaciais, ou de outra natureza, existentes entre os elementos primitivos, como na representação habitual por grafos de acordo com o paradigma estrutural.

Porém, ampliamos essa representação através da associação de cada vértice a um conjunto de variáveis aleatórias, que correspondem a características de aparência do respectivo elemento primitivo, e suas distribuições de probabilidade estimadas. Da mesma forma, associamos cada arco a um conjunto de variáveis aleatórias, que correspondem a características relacionais do respectivo par de elementos primitivos, e suas distribuições de probabilidade estimadas. Dessa forma, o GER corresponde a um modelo para uma classe de objetos estruturados, o qual descreve tanto características de aparência e relacionais, quanto estatísticas dessas propriedades, transmitindo noções sobre a variabilidade dos elementos primitivos e de suas relações.

Mediante essa representação, procedemos ao reconhecimento do conteúdo de uma imagem de entrada de acordo com os elementos modelados. O reconhecimento de objetos baseado em modelo equivale à determinação de uma rotulação das regiões da imagem que correspondem a um dado elemento modelado. Para utilizar as informações contidas no GER, extraímos da imagem de entrada medidas que equivalem a observações das variáveis aleatórias representadas por esse grafo modelo. Em seguida, criamos um grafo a partir dessa rotulação da imagem de entrada e das medidas calculadas, o qual denominamos *grafo de observação*.

Finalmente, formulamos o problema de reconhecimento sob uma ótica probabilística, em que avaliamos a hipótese de que a entrada seja, de fato, uma observação do modelo. Para tanto, investigamos duas formas de descrever essa probabilidade: a primeira, de acordo com a hipótese de independência estatística entre os elementos primitivos do modelo e suas correspondentes variáveis aleatórias; a segunda, de acordo com a hipótese de dependência condicional entre os elementos primitivos do modelo e suas correspondentes variáveis aleatórias.

Para ilustrar o interesse prático da metodologia desenvolvida, aplicamos seus conceitos ao domínio de imagens de ressonância magnética, visando a modelagem de órgãos abdominais e sua segmentação nesse tipo de imagem. A escolha desse domínio se justifica pela adequação da proposta da modelagem por GER às características dos órgãos abdominais: as diferentes fontes de variabilidade (anatômica, radiométrica, etc.) dos órgãos podem ser expressas através do aspecto estatístico do GER, enquanto o arranjo espacial e as relações desse sistema estruturado de órgãos são descritas através da topologia do grafo.

1.3 ORGANIZAÇÃO DO TEXTO

O Capítulo 2 apresenta um levantamento do estado-da-arte em reconhecimento estrutural de padrões, com enfoque sobre os métodos baseados em grafos e que procuram associar essa vertente com a abordagem estatística para classificação de objetos em visão computacional. O Capítulo 3 descreve a metodologia híbrida que desenvolvemos durante o doutorado. Nele, definimos a modelagem por grafo estatístico-relacional e descrevemos a interpretação probabilística para o reconhecimento de objetos baseado nesse modelo. Em seguida, o Capítulo 4 apresenta a aplicação da metodologia ao domínio de imagens de ressonância magnética abdominais, e a avaliação experimental realizada nesse âmbito. Finalmente, o Capítulo 5 traz um breve recapitulativo da metodologia, bem como conclusões sobre seus resultados e perspectivas futuras do trabalho.

2 REVISÃO BIBLIOGRÁFICA

As coisas têm peso, massa, volume, tamanho, tempo, forma, cor, posição, textura, duração, densidade, cheiro, valor, consistência, profundidade, contorno, temperatura, função, aparência, preço, destino, idade, sentido. As coisas não têm paz.

— Trecho da letra de *As coisas*, de Arnaldo Antunes

Neste capítulo, discutiremos alguns trabalhos que versam sobre o reconhecimento de objetos representados por grafos, dentro do paradigma estrutural, mas com tratamento estatístico e probabilístico para realizar a classificação de objetos almejada.

2.1 DEFINIÇÕES PRELIMINARES

Em primeiro lugar, nesta seção definiremos alguns conceitos importantes de teoria dos grafos e outros de grafos aplicados à visão computacional. Nas seções seguintes, discorreremos sobre os métodos do estado-da-arte que fazem uso de grafos.

2.1.1 *Conceitos elementares de teoria dos grafos*

Um *grafo* G é uma estrutura de dados denotada pelo par ordenado $G = (V, E)$, composto por um conjunto V de vértices, também denominados nós, e um conjunto $E \subseteq V \times V$ de arestas. Se V e E são conjuntos finitos, G constitui um grafo *finito*. As notações $|V|$ e $|E|$ serão adotadas para indicar a cardinalidade dos conjuntos V e E respectivamente. $|V|$ também é denominado *ordem* de G, enquanto $|E|$ é denominado *tamanho* de G.

Cada aresta de G está associada a um par de vértices do grafo, sendo denotada por $a = (v, w)$, com v e $w \in V$. O grafo é dito *dirigido*, ou *dígrafo*, se o par (v, w) é ordenado, com v representando a ponta inicial da aresta, enquanto w representa sua ponta final. Uma aresta de um grafo dirigido também pode ser chamada de *arco*. Quando v e w são coincidentes, a aresta é denominada *laço*. O dígrafo G é dito *simétrico* se, para cada aresta da forma (v, w), existe uma aresta da forma (w, v), $v, w \in V$. No outro extremo, um grafo *não-dirigido* é aquele em que as arestas consistem em pares não-ordenados de vértices, ou seja, se $v, w \in V$, $v \neq w$, e $(v, w) \in E$, então $(w, v) = (v, w) \in E$.

Dois vértices v e w de um dígrafo são *adjacentes*, ou vizinhos, se, e somente se, são extremidades de um arco de G, ou seja, se $(v, w) \in$

E ou $(w, v) \in E$. Para grafos não-dirigidos, tal relação é simétrica, pois $(v, w) = (w, v) \in E$ por definição. O conjunto N_E de vértices adjacentes ou vizinhos a um vértice $v \in V$ é definido como $N_E(v) = \{w_i \in V : (v, w_i) \in E \text{ ou } (w_i, v) \in E\}$. Se os vértices de G são dois a dois adjacentes, então o grafo é dito *completo*.

Um *caminho* P_G num grafo dirigido é uma sequência finita e não vazia de vértices distintos e arestas $P_G = (v_0, a_1, v_1, a_2, \ldots, a_k, v_k)$, em que $a_i = (v_{i-1}, v_i) \in E, \forall i = 1, \ldots, k$, $k \leqslant |V|$, de comprimento k. Um caminho entre dois vértices v e w é *mínimo* se não existe caminho de comprimento menor entre os mesmos vértices. Se existe ainda um arco $a = (v_k, v_0) \in E$, então P_G somado a a constitui um *ciclo* de G. No caso de grafos não-dirigidos, a noção de ciclo é válida quando $k \geqslant 3$.

Um grafo G é *conexo* se, para cada par de vértices distintos $v, w \in V$, existe um caminho ligando v a w. Se esta condição não for observada, G é considerado *desconexo*. Um grafo é *acíclico* se não apresenta ciclos em sua estrutura. Nesse caso, é também chamado *floresta*. Uma floresta conexa é denominada *árvore*.

A distância d entre dois vértices v e w, denotada $d_G(v, w)$, é definida como o comprimento do caminho mínimo existente entre ambos. Há também a noção de vizinhos de ordem-k de um vértice $v \in V$, os quais formam o conjunto de vértices w_i que distam k arcos de v, ou seja, $d_G(v, w_i) = k$. Se não existe caminho entre v e w, pode-se dizer que a distância entre ambos é infinita.

Um *subgrafo* $G' = (V', E')$ de $G = (V, E)$, ou $G' \subseteq G$, é um grafo tal que $V' \subseteq V$ e $E' \subseteq E$ e, para cada aresta $a \in E'$, seus extremos em V' são também extremos em V. Por sua vez, um *clique* é um subgrafo completo de G, ou seja, um subgrafo cujos vértices são dois a dois adjacentes. Um *clique maximal* é tal que nenhum outro clique o inclui. Finalmente, uma *componente* é um subgrafo conexo maximal de G, ou seja, a inclusão de outros vértices ou arestas em tal subgrafo viola a condição de conectividade do mesmo.

Nas próximas seções e demais capítulos, os grafos apresentados são todos finitos, podendo ser dirigidos ou não.

2.1.2 *Grafos e representação de dados pictóricos*

Como mencionado no capítulo introdutório, reconhecimento de padrões em visão computacional abrange tanto a representação de um padrão quanto a classificação do mesmo. Para a abordagem de reconhecimento estrutural, a representação do padrão se dá por meio de gramáticas ou grafos, sendo essa segunda forma a de interesse para o presente trabalho.

De particular interesse é o grafo relacional com atributos (GRA) (ou ARG, do inglês *attributed relational graph*), uma extensão do conceito de grafo, cujos vértices e arestas são associados a vetores de atributos

contínuos ou discretos, que exprimem um conjunto de características dos elementos que representam, ou de relações entre eles.

O trabalho de Tsai e Fu [TF79] foi um dos primeiros a definir essa estrutura no contexto de reconhecimento de padrões e, usualmente, ela é definida como uma tupla $G = (V, E, L_V, L_E)$, em que V e E expressam os mesmos conceitos apresentados anteriormente, enquanto $L_V : V \to \Sigma_V$ e $L_E : E \to \Sigma_E$ são funções de interpretação, respectivamente, dos vértices e arestas do grafo, onde Σ_V e Σ_E são os conjuntos de rótulos ou domínios de atributos possíveis associados aos vértices e arestas num dado problema.

Em reconhecimento de objetos representados por grafos, os vértices usualmente referem-se a elementos pictóricos da imagem, como um único pixel, um subconjunto de pixels (regiões) que compartilham propriedades similares (por ex., intensidade), formas geométricas específicas e até objetos inteiros. Já o conjunto de arestas do grafo pode representar inúmeros tipos de relações, como adjacências entre pixels ou regiões, posicionamento relativo no espaço (ao lado de, entre, acima, abaixo, etc.), pertinência (um objeto está contido em outro) e quaisquer outras relações possíveis.

Embora essa definição seja bastante genérica, diferentes trabalhos apresentam definições levemente diferenciadas de GRA, principalmente em relação à semântica associada aos conjuntos de atributos. Quando pertinente, essas definições específicas serão explicitadas.

No restante do texto, a seguinte nomenclatura será adotada para definir instâncias de grafos no problema de reconhecimento de padrões, ou objetos, em imagens:

- *grafo de entrada*: designa um grafo obtido a partir de dados de entrada, tipicamente, uma imagem 2D ou 3D;
- *grafo modelo* ou *protótipo*: designa um grafo representante de um ideal de padrão ou objeto, utilizado como referência no processo de classificação de um grafo de entrada representante de padrão desconhecido.

RECONHECIMENTO DE OBJETOS E GRAFOS. Dentro da representação de padrões por grafos, o problema de reconhecimento é interpretado como a busca por um mapeamento entre o conjunto de vértices do grafo de entrada e aquele do grafo modelo, ao que se denomina *casamento entre grafos*. Mais formalmente, o casamento entre um grafo $G_1 = (V_1, E_1)$ e outro $G_2 = (V_2, E_2)$ corresponde a uma função de mapeamento $f : V_1 \to V_2$.

Duas formas de abordar essa questão aparecem na literatura: métodos exatos e métodos inexatos. A primeira categoria impõe restrições rígidas quanto ao mapeamento e a preservação de estrutura, permitindo, principalmente, isomorfismos, isomorfismos de subgrafos, e homomorfismos. Na prática, esses métodos são pouco viáveis ou res-

tritos a situações específicas, já que modelo e entrada costumam apresentar discrepâncias. A categoria inexata pretende, assim, ser mais flexível e permitir encontrar uma solução sempre, através da atribuição de custos a cada mapeamento possível. Evidentemente, inúmeros mapeamentos existem, porém, nem todos são desejáveis. Para evitar soluções inadequadas, heurísticas e restrições de naturezas distintas são adotadas. Para um extenso estudo sobre as técnicas de casamento entre grafos existentes em reconhecimento de padrões e visão computacional, o leitor deve se referir ao artigo [CFSV04]. Essa etapa fundamental para o reconhecimento de um objeto constitui, ainda, um dos limitantes da abordagem estrutural, pois seu consumo de tempo é, na maior parte das técnicas, bastante elevado.

2.2 ABORDAGENS HÍBRIDAS PARA RECONHECIMENTO DE OBJETOS

Uma interpretação probabilística natural para o problema de reconhecimento de objetos baseados em grafos pode ser derivada da seguinte maneira: um grafo *protótipo* ou *modelo* representa uma classe de padrões ou objetos similares, enquanto um grafo de entrada gerado a partir de uma imagem pode retratar uma *realização*, ou *observação* de tal classe. Assim, um grafo modelo deve ser capaz, idealmente, de sintetizar a variabilidade existente numa mesma classe de objetos como se fosse um gerador dos padrões por ela caracterizados. Por outro lado, um grafo de entrada qualquer pertencerá a uma determinada classe de objetos de acordo com a probabilidade de ter sido gerado por um dado grafo modelo.

Essa interpretação permite a adoção de métodos estatísticos e probabilísticos em diferentes etapas do processo de reconhecimento de objetos representados por grafos:

- definição das características discriminantes dos objetos a serem reconhecidos e que serão utilizadas tipicamente como atributos associados a vértices e arestas do grafo;
- criação ou aprendizagem do grafo modelo;
- processo de casamento entre grafos, para determinar a correspondência entre modelo e entrada.

Esses três aspectos podem ser usados para definir uma taxonomia para classificação dos trabalhos mais expressivos existentes na literatura e que buscam uma abordagem híbrida entre os reconhecimentos estrutural e estatístico. Em particular, os métodos híbridos existentes relativos à etapa de seleção de características (ou atributos) consistem simplesmente na aplicação de técnicas de seleção de características, como aquelas discutidas em [DHS00, JZ97, JDM00]. Nas próximas subseções, portanto, serão discutidas apenas as técnicas que adotam

uma interpretação estatística para a geração do modelo e para a etapa de classificação por casamento entre grafos.

2.2.1 *Métodos híbridos e geração de modelo*

Uma classe de padrões ou objetos pode ser descrita em função de propriedades que a caracterizam, levando-se em consideração a variabilidade entre as suas diferentes expressões. Dentro da interpretação probabilística para reconhecimento estrutural de objetos representados por grafos, uma classe pode ser representada por um grafo modelo ou protótipo, construído a partir de amostras, e com o qual os dados desconhecidos, representados por grafos de entrada, são comparados.

GRAFOS ALEATÓRIOS. Diferentes construções para tal grafo modelo são apresentadas na literatura. Um conceito inicial relevante é o de *grafos aleatórios* [Mar04], os quais aproximam os domínios de teoria dos grafos e de probabilidade. Um grafo aleatório é formado por um processo aleatório, tipicamente constituído pela adição de arestas aleatórias entre os vértices que o compõem. Os processos aleatórios que originam um tal grafo determinam diferentes distribuições de probabilidade sobre os grafos gerados. Esse conceito compreende, portanto, apenas a variabilidade estrutural possível para conjuntos de grafos com um mesmo número de vértices.

Uma extensão desses grafos aleatórios para grafos relacionais com atributos aplicados ao reconhecimento estrutural de padrões foi proposto por Wong e Shen [WS90]. Essa extensão trouxe a definição de *grafos aleatórios genéricos* (GRG, do inglês, *generalized random graphs*), os quais associavam os atributos de vértices e arestas a variáveis aleatórias seguindo uma distribuição de probabilidade conjunta. Embora essa formulação seja extremamente poderosa, ela é intratável do ponto de vista computacional devido à complexidade da estimativa das probabilidades conjuntas.

Algumas simplificações sobre essa definição foram propostas. Wong e You [WY85] impuseram uma forte suposição de independência entre vértices, entre arestas, e entre vértices e arestas não conectados, tornando o método praticável, mas muito generalista para a criação de modelos de classes que preservem efetivamente características importantes.

GRAFOS DESCRITOS POR FUNÇÕES. Em busca de uma definição mais robusta e ainda praticável, foi introduzido o conceito de *grafo descrito por funções* (do inglês, *function-described graph* ou FDG) [SAS03]. Essa estrutura foi utilizada para sintetizar, num modelo, informações presentes numa série de GRA num conjunto de referência, e que representam, em particular, diferentes faces de um objeto tridimensional.

Tal modelo é criado de forma a aceitar GRA que façam parte da classe do objeto e rejeitar os que a ela não pertençam, conforme esses obedeçam ou violem certas regras estruturais representadas por funções, além de ser utilizado como gerador de grafos de *realização* de um FDG.

Informalmente, um FDG é representado por seu conjunto de vértices aleatórios e de arestas aleatórias, aos quais são associadas variáveis aleatórias definidas sobre um dado domínio de atributos de vértices ou de arestas. Vértices ou arestas ditas nulas também podem ser criadas no modelo para representar elementos que não estejam presentes numa face do objeto, mas que sejam visíveis em outra. A viabilidade do modelo na prática é atribuída à suposição de independência dos atributos entre vértices e entre arestas, mas limitando a existência de arestas não-nulas, as quais devem sempre conectar dois vértices também não-nulos. Já a robustez do modelo é associada às funções de antagonismo, ocorrência e existência definidas sobre pares de vértices, as quais representam, respectivamente, elementos que nunca aparecem simultaneamente num GRA do conjunto de referência, elementos que sempre aparecem simultaneamente, e a existência de pelo menos um dos elementos da classe representada pelo FDG em um dos GRA de referência.

Para realizar o reconhecimento de um GRA em relação a um FDG, os autores propuseram um algoritmo sub-ótimo para determinar o casamento inexato entre ambos os grafos. O algoritmo chamado *eficiente* realiza essa tarefa através da minimização de uma medida de dissimilaridade entre os grafos que é uma aproximação da máxima probabilidade *a posteriori* de uma função de rotulação de vértices. Um fator desfavorável dessa formulação por FDG é a complexidade de espaço do FDG, que aumenta exponencialmente com o número de relações a serem consideradas (e.g.: relação entre vértices versus relação entre vértices e arestas).

GRAFOS RELACIONAIS GENERALIZADOS COM ATRIBUTOS. Outra
proposta para modelar conjuntos de GRA foi introduzida em [FGV00]: grafos relacionais generalizados com atributos (GARG, do inglês *generalized attributed relational graphs*). Um GARG é um GRA cujos conjuntos de tipos possíveis para os atributos de vértices e arestas compreendem um valor especial ϕ, que pode ser usado como um "coringa", ou seja, é compatível com qualquer tipo de atributo e sem parâmetros próprios. Nesta metodologia, a partir de um conjunto de GRA previamente rotulados, pertencentes a uma única classe de objetos ou não, é possível derivar automaticamente grafo relacional genérico com atributos (*generalized attributed relational graph*) (GARG) protótipos que os representem. Para determinar esses modelos, o algoritmo parte de protótipos triviais (por ex.: um grafo com um único vértice) e os incrementa até que se tornem consistentes e completos. Esses

dois conceitos referem-se, respectivamente, à existência de um GARG gerador para todo grafo do conjunto de referência e a certeza de que, se um GARG pode gerar um dado GRA de referência, então a classe ou rótulo de ambos é a mesma. Para avaliar o primeiro aspecto, uma medida de entropia relativa à representação da classe em bits foi utilizada, enquanto o segundo aspecto considerou apenas o número de GRA de referência que podem ser gerados por um certo GARG. O método foi aplicado sobre um banco de imagens de dígitos e apresentou resultados encorajantes. Porém, o tempo levado para construção dos conjuntos de protótipos é ainda uma questão que penaliza a viabilidade da técnica para conjuntos de padrões mais complexos.

MODELOS DE ESTRUTURAS PICTÓRICAS. Em [FH05], os autores propõem um método estatístico para gerar modelos de objetos representados por partes e seu respectivo arranjo deformável, chamados de *estruturas pictóricas*, a partir de exemplos. O método proposto permite não apenas aprender modelos, mas também reconhecer instâncias de um objeto presentes numa imagem. Esse último aspecto, porém, será discutido na subseção seguinte. Uma estrutura pictórica é descrita por um grafo modelo não-dirigido em que os vértices correspondem às partes do objeto representado, enquanto as arestas conectam certos pares de partes relacionados. Uma instância de um objeto é dado por uma configuração do conjunto de dados das localizações (como posição, orientação) das partes do mesmo.

Dado um conjunto de m imagens de referência (ou treinamento) de um objeto e suas respectivas configurações de localização, deseja-se obter uma estimativa dos parâmetros $\theta = (u, E, c)$ que caracterizam um modelo de estrutura pictórica, em que u se refere aos parâmetros de aparência de cada uma das partes, E é o conjunto de conexões entre partes e c é um conjunto de parâmetros das conexões presentes em E. Para obter θ, é utilizado o método de máxima verossimilhança para estimar tanto a aparência, quanto as dependências entre as partes. No entanto, os arranjos deformáveis permitidos entre as partes restringem-se àqueles que obedecem uma condição de distância entre as localizações originais das partes e uma transformada dessas localizações para um novo espaço discretizado, que resultem em grafos acíclicos (árvores), para fins de otimização da desempenho computacional. O método foi aplicado aos problemas de detecção de faces e de articulações do corpo humano em imagens 2D, com resultados bastante satisfatórios.

2.2.2 *Métodos híbridos e casamento entre grafos*

Conforme mencionado na subseção anterior, o método discutido em [FH05] também permite o reconhecimento de objetos em imagens e não apenas a criação de modelos. Dentro do arcabouço estatístico, o problema

de casamento entre um modelo de estrutura pictórica e uma imagem de entrada pode ser entendido como a busca da localização mais provável do objeto numa dada imagem, o que é realizado através da estimação da máxima probabilidade *a posteriori* (MAP) de tal localização. Apesar dessa estimativa representar um problema NP-difícil para grafos e funções arbitrárias, os autores sustentam que as restrições impostas sobre o modelo (limites sobre as distâncias entre partes) permitem torná-lo mais eficiente para problemas práticos. Para casos em que não se deseja encontrar apenas a melhor localização estimada do objeto na imagem, mas sim um conjunto de localizações prováveis, os autores propõem também a realização de amostragem a partir da distribuição *a posteriori* do objeto, dados os parâmetros definidos para seu modelo e uma imagem de entrada.

2.3 TENDÊNCIAS NO ESTUDO DE CASAMENTO ENTRE GRAFOS

O artigo de Bunke et al. [BIN05] analisa o problema de casamento entre grafos para reconhecimento de padrões, sugerindo tendências e caminhos para aprimorar as técnicas existentes na literatura ou desenvolver novos métodos. Em particular, os autores propõem o estudo de equivalências entre métodos recentes de reconhecimento estatístico de padrões e métodos estruturais, com destaque para as seguintes abordagens: sistemas de multi-classificadores, aprendizagem de custos para algoritmos de edição de grafos.

Algoritmos para edição de grafos consistem em procedimentos de transformação de um grafo em outro, de acordo com operações como inserção, remoção, ou substituição de vértices e/o arestas. De modo geral, custos são associados a cada tipo de operação, de acordo com sua importância num dado problema. A distância de edição entre dois grafos permite comparar o grau de compatibilidade ou semelhança entre ambos e é usualmente definida como o mínimo custo calculado dentre todas as sequências de operações que transformam um grafo no outro. Essa medida pode ser utilizada para classificar um grafo de entrada em relação a diversos grafos modelos, por exemplo. Assim, a determinação de custos adequados para as operações elementares é primordial na obtenção de uma distância de edição satisfatória. Em busca de tal objetivo, os autores sugerem a proposição de técnicas automáticas de aprendizagem computacional desses custos, discutindo um método em particular baseado em mapas organizáveis, criados a partir de um conjunto de grafos de referência.

Já a idéia dos multi-classificadores surgiu em reconhecimento estatístico de padrões e consiste na composição das saídas de classificadores distintos, para determinar um classificador geral supostamente mais robusto. Para a temática dos métodos híbridos, os autores apontam a criação de sistemas multi-classificadores que agregam tanto classificadores estatísticos quanto classificadores estruturais, já que

ambos costumam ser suficientemente distintos em sua natureza, podendo trazer benefícios ao processo de classificação se reconhecerem a entrada sob critérios complementares.

No próximo capítulo, discutiremos uma proposta híbrida que incorpora estatísticas e probabilidade à modelagem por grafos relacionais com atributos e, igualmente, à etapa de casamento entre grafos.

3 MODELAGEM E RECONHECIMENTO DE OBJETOS ESTRUTURADOS

Os olhos, eles mesmos, são estúpidos. Eles não têm o poder para discriminar as coisas dignas de serem vistas das coisas não dignas de serem vistas. (...) A capacidade de discriminar não pertence aos olhos. Pertence ao olhar.

— Rubem Alves em *Sobre Ciência e Sapiência*

Um *objeto estruturado* é aquele cuja integralidade pode ser descrita através de elementos primitivos que o compõem. Retomando os exemplos apresentados no Capítulo 1, são objetos estruturados: veleiro, composto por barco, vela, e mastro; avião, composto por suas asas direita e esquerda, cockpit, e fuselagem.

Da mesma forma, um *sistema estruturado* de objetos é aquele que pode ser descrito segundo os objetos (elementos primitivos) que o compõem. Alguns exemplos de sistemas estruturados são: rosto, composto por lábios, nariz, olhos, sobrancelhas, pele; corpo humano, subdividido em cabeça, tronco, braços e pés direito e esquerdo, mãos e pernas direita e esquerda.

Todos esses objetos ou sistemas apresentam características de aparência e, sobretudo, relacionais, que são preservadas entre indivíduos distintos dessas classes, ainda que haja um certo grau de variabilidade entre eles. A estrutura típica de cada objeto, bem como as relações existentes entre os elementos primitivos, definem uma rede semântica [GW93], que configura um modelo conceitual do objeto estruturado em questão. Essa rede semântica pode ser expressa computacionalmente através de um grafo relacional com atributos, tipicamente adotado no paradigma de reconhecimento estrutural de padrões, em que nós representam os elementos primitivos, e arcos exprimem as relações entre eles. Já a variabilidade intrínseca aos elementos primitivos e às suas relações pode ser modelada através de aprendizagem supervisionada de atributos que os descrevem e métodos estatísticos, como propõe a abordagem de reconhecimento estatístico de padrões.

Dessa forma, a modelagem de objetos ou sistemas estruturados apresenta um caráter híbrido que se enquadra diretamente no contexto de pesquisa mencionado no Capítulo 1, ou seja, a investigação de maneiras de combinar os métodos estatísticos de reconhecimento de padrões àqueles de reconhecimento estrutural de padrões, para usufruir dos benefícios e conhecimentos provenientes de ambas as abordagens.

Portanto, neste capítulo, descrevemos a proposta estatístico-estrutural que desenvolvemos durante esta tese de doutorado, para modelar

objetos estruturados através de uma representação híbrida, e utilizar esse tipo de modelo para reconhecer um dado objeto estruturado numa imagem.

3.1 DEFINIÇÕES E NOTAÇÕES

No restante do presente capítulo, adotaremos os conceitos e notações descritos nesta seção. Considere a representação digital P de uma imagem. O domínio de P é uma região do espaço dada pelo seguinte conjunto de elementos pictóricos, se P é bidimensional:

$$R = \{(x, y) 0 \leqslant x < h, 0 \leqslant y < w, (w, h) \in \mathbb{N}_+^2\}, \tag{3.1}$$

em que w é o número de colunas da imagem, e h é seu número de linhas.

Se P é uma imagem tridimensional, então seu domínio é dado por:

$$R = \{(x, y, z), 0 \leqslant x < h, 0 \leqslant y < w, 0 \leqslant z < d, (w, h, d) \in \mathbb{N}_+^3\}, \tag{3.2}$$

em que w e h são como no caso bidimensional, e d é sua profundidade.

Denotaremos a função de intensidade associada a um elemento pictórico (pixel ou voxel) de P por $I : R \mapsto \mathbb{R}^+$, i.e., $I(x, y)$ ($I(x, y, z)$) é o valor da intensidade do elemento pictórico dado pelas suas respectivas coordenadas. O número total de elementos pictóricos da imagem é dado por $|R| = w \times h$ ($|R| = w \times h \times z$).

Seja L o conjunto de rótulos representantes dos elementos primitivos de um objeto estruturado (ou sistema estruturado de objetos) a serem modelados, e ρ, um rótulo associado a todo o universo não representado por L. Considere também $L^* = L \cup \{\rho\}$.

Uma segmentação completa [SHB08] da imagem P, baseada num modelo representando L^*, consiste numa partição S de R tal que $S = \{s_1, \ldots, s_{max}\}$, $max \leqslant |R|$, associada a um mapeamento $f : S \mapsto L^*$. Cada s_i, também denominado *região refinada*, corresponde a um subconjunto de pontos pertencentes a R. Esses conjuntos obedecem às seguintes propriedades [GW93]:

1. $\bigcup_{i=0}^{max} s_i = R$;
2. cada s_i é uma região conexa;
3. $s_i \cap s_j = \emptyset, \forall i \neq j$;
4. $f(s_i) \in L^*, \forall i = 1, \ldots, max$.

Note que, num caso extremo, S corresponde ao conjunto de subconjuntos unitários de R, ou seja, cada elemento pictórico da imagem corresponde a uma região s_i. Outro caso extremo é aquele em que toda a imagem é considerada como uma única região ($S = R$).

Definiremos ainda um conjunto s_l, referente a cada elemento $l \in L^*$, tal que $s_l = \{s_i \subset S : f(s_i) = l\}$, ou seja, s_l é um conjunto de regiões disjuntas às quais um mesmo rótulo foi atribuído. O conjunto s_l também será denominado conjunto das *regiões expandidas*. Denotaremos ainda $|s_i|$ o número de voxels pertencentes a s_i, e $|s_l|$ o número total de regiões associadas a l em S.

Uma única imagem P pode ser associada a diferentes segmentações. Quando houver necessidade de nos referirmos a segmentações distintas de uma mesma imagem, utilizaremos a seguinte notação: S^j, para a partição referente à j–ésima segmentação de P ($j = 1, \ldots, max^j$, $max^j = (2^{|R|})^{|L^*|}$); f^j, para o respectivo mapeamento; s^j_l, para o conjunto de regiões de S^j rotuladas como pertinentes ao órgão l.

Ainda, para nos referirmos às segmentações correspondentes a diferentes imagens $P_1, P_2, \ldots$, adotaremos a notação a seguir: I_k, para a função de intensidade referente à k–ésima imagem P_k ($k = 1, \ldots$); R_k, para o domínio de P_K; S_k, para uma partição de R_K; f_k, para o seu respectivo mapeamento; s_{k_l}, para o conjunto de regiões de S_k rotuladas como pertinentes ao elemento primitivo l.

3.2 MODELAGEM DE OBJETOS ESTRUTURADOS

3.2.1 *Modelo conceitual*

Um modelo conceitual de um objeto ou sistema estruturado consiste numa descrição do conhecimento existente acerca desse objeto, considerando as limitações decorrentes da representação pictórica digital do mesmo. Para reunir os conhecimentos necessários para definir um objeto de acordo com os paradigmas estrutural e estatístico, o modelo conceitual deve fornecer as seguintes informações:

- quais são elementos primitivos que compõem o objeto ou sistema de interesse, o que define o conjunto $|L^*|$;
- qual é o arranjo estrutural dos elementos primitivos;
- quais são as características de aparência que podem ser extraídas dos elementos primitivos;
- quais são as características relacionais que podem ser extraídas dos elementos primitivos, respeitando o arranjo estrutural.

Para responder a essas perguntas, é preciso considerar uma série de aspectos, dentre os quais, o quão discriminantes e robustos são os descritores de aparência e relacionais, e o quão facilmente eles podem

ser observados em imagens contendo um dado objeto. Portanto, a definição do modelo conceitual é um passo crítico da metodologia.

Os métodos de seleção de características [JZ97, DHS00] oriundos de reconhecimento estatístico de padrões propõem soluções eficientes para tratar essas questões quando as características de aparência e relacionais são facilmente quantificáveis, e se resumem a um valor numérico. Essa observação é relevante sobretudo em termos das características relacionais, já que alguns descritores não se resumem a um número, como é o caso de relações espaciais (por ex. "A à direita de B"), que podem determinar toda uma região do espaço em que essa relação é válida entre dois objetos. Nesses casos, uma solução é substituir a relação por alguma medida mais simples que a represente.

Para ilustrar o modelo conceitual, retomemos o exemplo da representação de um avião apresentado no Capítulo 1. Um modelo conceitual possível para esse objeto pode ser assim definido:

- elementos primitivos: *cockpit*, asa direita, asa esquerda, fuselagem;
- arranjo estrutural: o *cockpit* está à frente da fuselagem, a asa direita está à direita da fuselagem, e a asa esquerda está à esquerda da fuselagem;
- características de aparência: intensidade, tamanho;
- características relacionais: contraste, proporção, ângulo entre os eixos principais de cada elemento primitivo.

É importante observar que o modelo conceitual permite descrever um objeto de maneira bastante detalhada. Porém, é desejável que ele seja o mais conciso possível, para que sua representação computacional não seja excessivamente complexa.

3.2.2 *Modelo computacional: grafo estatístico-relacional*

Mediante o modelo conceitual, transferimos sua descrição para uma única estrutura de dados, que denominamos *grafo estatístico-relacional* (GER, ou SRG, para o inglês *statistical relational graph*[1]).

Formalmente, definimos o grafo estatístico-relacional como uma quádrupla $G_m = (V_m, E_m, \hat{\theta}^{obj}, \hat{\theta}^{rel})$ em que:

- V_m é o conjunto de nós do grafo G_m. $|V_m|$ representa a ordem de G_m e $|V_m| = |L^*|$. Cada nó representa um único elemento primitivo pertencente a L^* e pode ser denotado por v_m, ou v_m^i, $i = 1, \ldots, |V_m|$, segundo a necessidade;

1 Essa mesma nomenclatura foi adotada no artigo [CH93]; entretanto, a definição do grafo não corresponde àquela proposta nesta tese.

- E_m é o conjunto de arcos do grafo G_m. $|E_m|$ representa o tamanho de G_m. Cada arco representa as relações existentes entre elementos primitivos pertencentes a L^*. Denotamos um arco entre os nós $v_m^i \in V_m$ e $v_m^j \in V_m$ por $e_m^{ij} = (v_m^i, v_m^j)$, ou, simplesmente por e_m, quando os nós conectados não forem especificados;

- $\hat{\theta}^{obj}(\overrightarrow{X}(v_m)) = [\hat{\theta}(X_1(v_m)), \ldots, \hat{\theta}(X_\nu(v_m))]$, $v_m \in V_m$ é um mapeamento do vetor de variáveis aleatórias $\overrightarrow{X}$ associadas a um nó v_m para um vetor de parâmetros estimados das distribuições referentes às variáveis aleatórias que representam as propriedades do elemento primitivo modelado pelo nó v_m. O símbolo ν_{V_m} denota o número de propriedades de aparência consideradas para esse elemento primitivo;

- $\hat{\theta}^{rel}(\overrightarrow{Y}(e_m)) = [\hat{\theta}(Y_1(e_m)), \ldots, \hat{\theta}(Y_{\nu_{E_m}}(e_m))]$, $e_m \in E_m$ é um mapeamento do vetor de variáveis aleatórias $\overrightarrow{Y}(e_m)$ de um arco e_m para um vetor de parâmetros estimados das distribuições referentes às variáveis aleatórias representantes das relações partilhadas entre os elementos primitivos modelados pelos vértices conectados pelo arco e_m. O símbolo ν_{E_m} denota o número de relações consideradas.

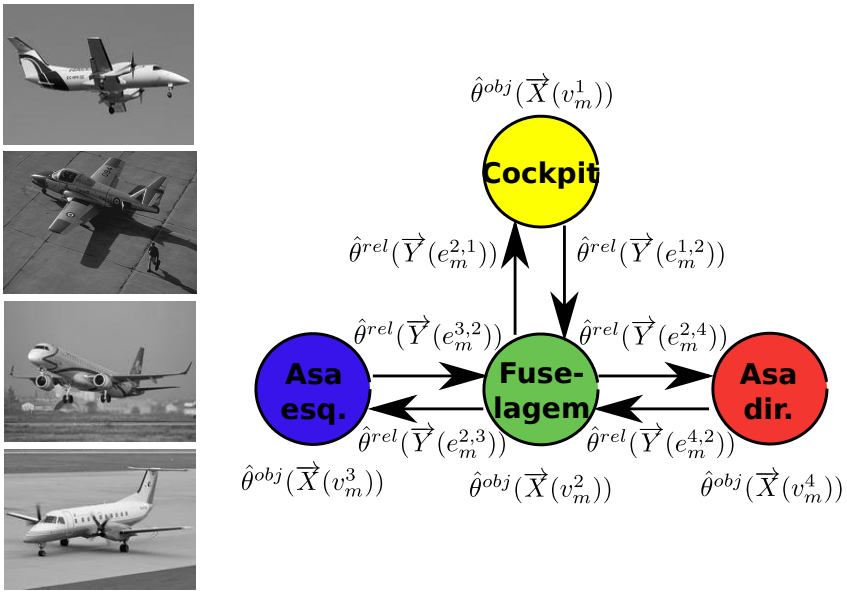


Figura 3: Representação de objetos por GER: avião e exemplo de respectivo GER. Cada nó do grafo está associado a um conjunto de variáveis aleatórias, representando atributos de aparência, e aos parâmetros de suas densidades de probabilidade. Da mesma forma, cada arco do grafo está associado a um conjunto de variáveis aleatórias, representando atributos relacionais, e aos parâmetros de suas densidades de probabilidade.

A Figura 3 representa um GER, correspondente ao avião apresentado no Capítulo 1. Cada nó representa um elemento primitivo, enquanto os arcos indicam o arranjo espacial desses elementos e a existência de relações entre eles. Finalmente, vetores de parâmetros referentes às variáveis aleatórias, que modelam as características de aparência e relacionais, são associados tanto aos nós quanto aos arcos do GER.

CRIAÇÃO DO GRAFO MODELO E ESCOLHA DOS ATRIBUTOS. Para calcular os vetores de parâmetros das características dos elementos primitivos e suas relações, a metodologia requer um conjunto de amostras de treinamento representativas da classe de objeto estruturado de interesse. Dessas imagens serão extraídos esses mesmos descritores e estimados os parâmetros para as suas distribuições de probabilidade associadas.

A observação das amostras disponíveis permite também auxiliar na escolha dos atributos e das relações através dos métodos de seleção de características, ou do conhecimento *a priori* fornecido por especialistas para uma certa aplicação.

Para extrair as medidas de interesse, as imagens de treinamento devem ser segmentadas de modo supervisionado manualmente, ou por um método semi-automático [NGCJ+12], em que os elementos primitivos, definidos no modelo conceitual, sejam rotulados. Essas rotulações serão consideradas ideais (*ground truth*).

De posse dessas informações, a criação do grafo modelo G_m segue a sequência de passos ilustrados na Figura 4, e apresentados no algoritmo da Figura 5. Seja N o total de imagens de treinamento disponíveis. O algoritmo recebe um vetor $P[1 \dots N]$, contendo as N imagens de treinamento de um dado objeto, e um segundo vetor $P'[1 \dots N]$ contendo as N imagens correspondentes às rotulações ideais das imagens de treinamento. Para criar o GER, também são fornecidos os seguintes parâmetros: o conjunto L^* e uma matriz de adjacências binária mtz_adj, com dimensões $|L^*|$ $times|L^*|$, para indicar os elementos primitivos que estão relacionados.

Para cada imagem de treinamento, o algoritmo primeiro calcula as características (COMPX) de aparência para cada elemento primitivo $l_i \in L^*$, a partir das regiões assim rotuladas na k-ésima imagem ideal e sua respectiva original. Esses dados são reservados num vetor de características para cada imagem e cada elemento primitivo (linhas 4-9). Da mesma forma, as características relacionais são calculadas (COMPY) para cada par de elementos primitivos $l_i \in L^*$ e $l_j \in L^*$, segundo a estrutura dada por mat_adj, e armazenados num vetor de características para cada imagem e cada par de elementos relacionados (linhas 10-18).

Em seguida, o grafo G_m pode ser gerado: os nós são criados para cada elemento primitivo modelado, e os parâmetros das caracterís-

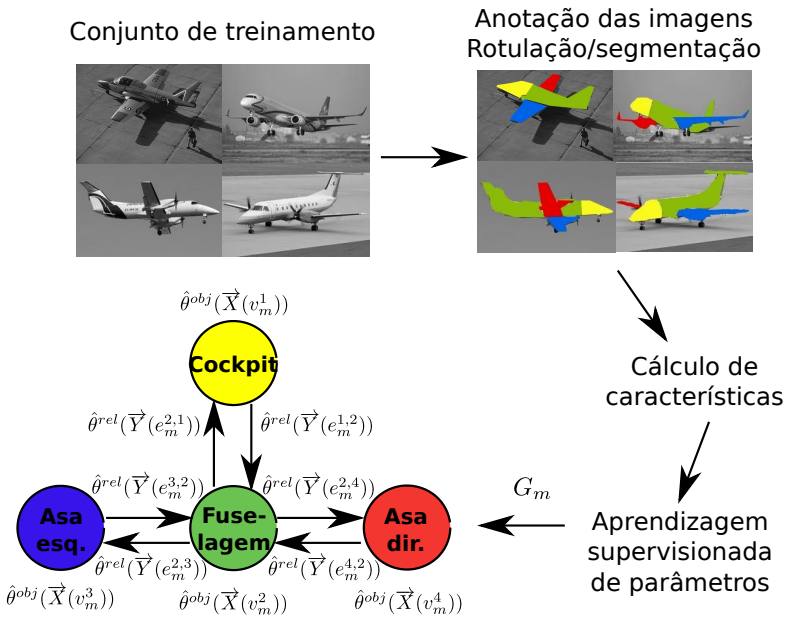


Figura 4: Passos para gerar um GER. A partir de um conjunto de imagens de treinamento, em que cada elemento primitivo foi devidamente rotulado de maneira supervisionada, calculamos medidas de aparência e relacionais com base em cada imagem e as regiões rotuladas. Essas medidas servirão para estimar os parâmetros das leis de probabilidade que regem as respectivas variáveis aleatórias. O grafo estatístico-relacional é então gerado em função desses elementos.

ticas de aparência que os descrevem são estimados (ESTIMAPARAMS) segundo os vetores de características armazenados anteriormente (linhas 20-24); do mesmo modo, os arcos são criados para cada par de elementos primitivos relacionados segundo mat_adj, e os parâmetros das características relacionais que os descrevem são estimados (ESTIMAPARAMS) (linhas 26-31).

A implementação dos procedimentos COMPX e COMPY depende das características escolhidas. Apresentaremos um exemplo completo do cálculo dessas características distintas no Capítulo 4, quando estudarmos um problema de modelagem de órgãos abdominais. Da mesma forma, o procedimento ESTIMAPARAMS) depende do método escolhido (por ex., estimador Bayesiano, ou estimador de máxima verossimilhança) para estimar os parâmetros das características, conforme as suposições estatísticas feitas para cada objeto estruturado modelado.

3.3 RECONHECIMENTO ESTATÍSTICO-ESTRUTURAL BASEADO EM MODELO

O modelo de um objeto estruturado introduz conhecimento *a priori* a respeito desse objeto, e pode ser considerado um protótipo dos

```
criaGER(P[1..n], P'[1..n], L*, mtz_adj)
// Cálculo de características para cada elemento primitivo
// em cada imagem de treinamento
for k = 1 to n
    // para cada elemento primitivo l_i rotulado em P'[k]
    for i = 1 to |L*|
        // para cada observação x_idx associada a v.a. X_idx
        // de l_i em P'[k]
        for idx = 1 to ν_{V_m}
            x_idx[l_i][k] = compX(l_i, idx, P[k], P'[k])
    // para cada par de elementos primitivos rotulados em P'[k]
    // e relacionados segundo mtz_adj
    for i = 1 to |L*|
        for j = 1 to |L*|
            // para cada observação y_idx associada a v.a. Y_idx
            // do par de elementos l_i, l_j em P'[k]
            for idx = 1 to ν_{E_m}
                if mtz_adj[i][j] == true
                    y_idx[l_i][l_j][k] = compY(l_i, l_j, idx, P[k], P'[k])
// Construção de G_m
// Criar nó associado ao elemento primitivo l_i e adicioná-lo a G_m
for i = 1 to |L*|
    v_m^i = Nó(l_i)
    V_m = V_m ∪ {v_m^i}
    θ̂^obj(X⃗(v_m^i)) = estimaParams(x[l_i][n])
// Criar um arco para cada par de objetos
// relacionados segundo mtz_adj
for i = 1 to |L*|
    for j = 1 to |L*|
        if mtz_adj[i][j] == true
            e_m^ij = Arco(l_i, l_j)
            E_m = E_m ∪ {e_m^ij}
            θ̂^rel(Y⃗(e_m^ij)) = estimaParams(y[l_i][l_j][n])
return G_m
```

Figura 5: Pseudo-código para a criação de um GER (G_m).

diferentes exemplares desse objeto. Por sua vez, o problema de reconhecimento baseado em modelo pretende considerar essas informações no processo de classificação do conteúdo de uma imagem, para nela reconhecer as regiões que correspondem aos elementos primitivos modelados. Portanto, no restante desta seção, discutiremos uma metodologia para reconhecer objetos estruturados representados por GER numa dada imagem.

3.3.1 *Da representação digital da imagem ao reconhecimento segundo modelo*

Uma imagem P é representada digitalmente de diferentes maneiras. A representação por grade, ou matricial, é baseada em elementos pictóricos (pixels ou voxels) que representam pontos do domínio R. Nessa representação, os elementos pictóricos são associados a uma função de intensidade I apenas, e a estrutura entre eles deriva da topologia da grade.

Outra representação de P é aquela dada por regiões. Nesse caso, o conteúdo da imagem P é representado por um conjunto de regiões correspondentes a uma partição S de P. Por agregar os elementos pictóricos da imagem, o uso de regiões conduz a uma representação mais compacta de P, que permite também o cálculo de características para cada s_i que complementam a intensidade das regiões, e de relações entre s_i e s_j distintos, tipicamente adjacentes.

Uma partição S pode ser obtida a partir de algoritmos de segmentação, tais como os algoritmos de linha de partição de águas (*watershed*) [VS91], ou *quad-trees* [SW85]. Nesses casos, cada s_i corresponde a uma região disjunta produzida por um tal algoritmo.

Por exemplo, numa segmentação por linha de partição de águas do gradiente de uma imagem P, cada s_i corresponde a uma bacia de captação de águas. Em geral, essa técnica produz uma supersegmentação da imagem de entrada, ou seja, resulta em diversas micro-regiões induzidas pelas intensidades da imagem e a configuração espacial dos elementos pictóricos. Embora a segmentação produza esse efeito, o algoritmo de *watershed* costuma ser bastante eficiente e a supersegmentação não impede a recuperação dos verdadeiros contornos dos objetos a partir das regiões que os compõem.

O reconhecimento de objetos estruturados em P, baseado no modelo estatístico-estrutural, equivale à rotulação de todos seus pixels (voxels) ou regiões de acordo com os elementos primitivos modelados a que correspondem. Apesar de haver um número exponencial de rotulações possíveis para P, apenas algumas são desejáveis, a saber, aquelas em que as regiões refinadas correspondem de fato a elementos primitivos do modelo.

Portanto, procuramos uma rotulação f para S que produza um conjunto de regiões expandidas s_l para cada elemento primitivo $l \in L^*$ presente em P. Além disso, as características de aparência e relacionais medidas a partir dessas regiões expandidas devem ser compatíveis com as informações estatísticas armazenadas no modelo.

Porém, pixels (voxels) e regiões provenientes de supersegmentações não pertencem à mesma escala de representação em que se encontram os elementos primitivos do modelo estatístico-relacional, como ilustrado na Figura 6. Logo, as características de aparência e, sobretudo, as relações provenientes dessas representações refinadas não

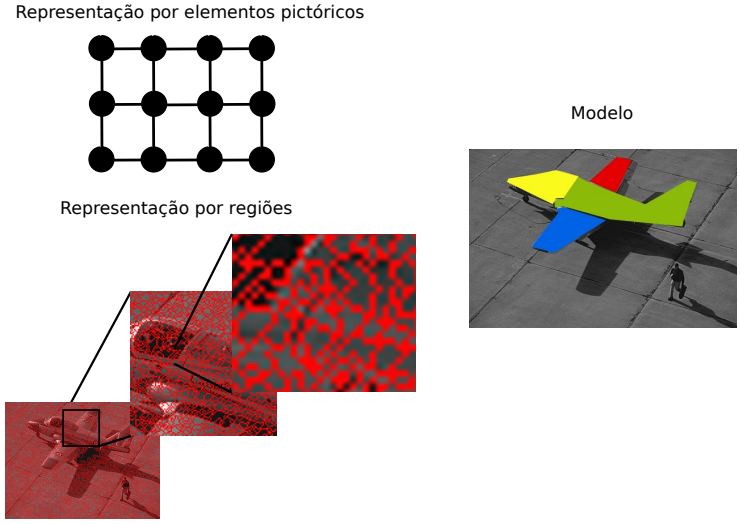


Figura 6: Representação da imagem através de elementos pictóricos (pixel/voxel) em grade (acima à esq.), através de regiões (abaixo à esq.) e representação do modelo (à dir.) que caracteriza a integralidade dos elementos primitivos. As características obtidas a partir dessas representações fazem parte de escalas diferentes e, portanto, não são diretamente comparáveis aos dados de alto nível contidos no modelo.

são diretamente comparáveis àquelas transmitidas pelo GER modelo, que se referem a elementos primitivos em sua integralidade.

Para passar da representação refinada à expandida e avaliar a compatibilidade da entrada em relação ao modelo, propusemos uma representação auxiliar, que denominaremos *grafo de observação*, definida a seguir.

3.3.2 *Grafo de observação*

Um grafo de observação é definido como um GRA, em que seus conjuntos de nós e de arcos representam, respectivamente, conjuntos de regiões expandidas e suas relações. Além disso, os vetores de atributos associados aos nós são um conjunto de observações tomadas de cada s_l, e correspondentes às variáveis aleatórias que descrevem características de aparência do modelo. De modo similar, os vetores de atributos associados aos arcos são um conjunto de observações tomadas de pares de regiões expandidas, correspondentes às variáveis aleatórias que descrevem características relacionais do modelo. A Figura 7 ilustra esse conceito.

As medidas que geram essas observações são extraídas a partir das propriedades de cada um dos conjuntos s_l de pixels (voxels) da imagem de entrada que correspondem a um único rótulo identificador de um elemento primitivo l no modelo. Logo, diferentes segmentações e rotulações produzem conjuntos distintos de pixels rotulados e,

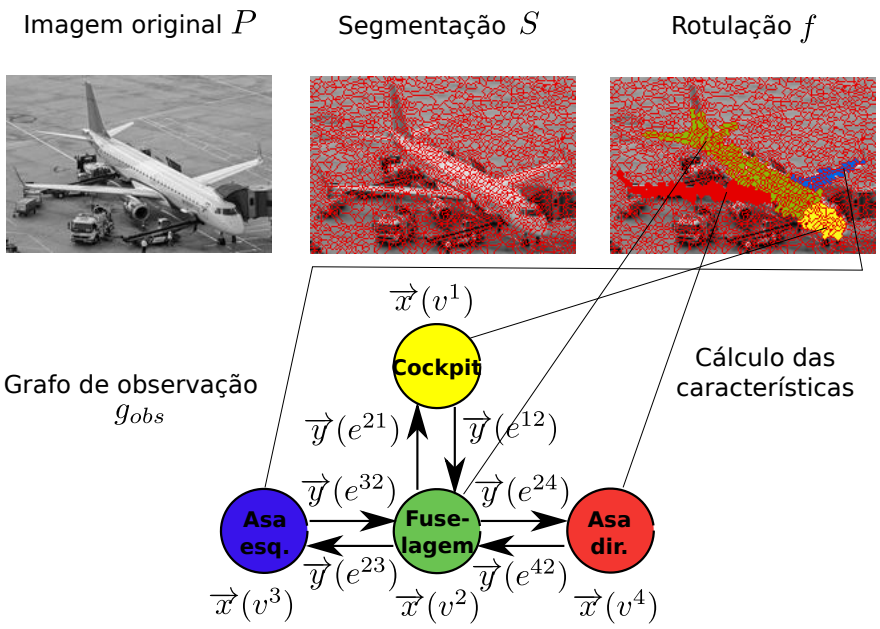


Figura 7: Exemplo de grafo de observação: dada uma imagem contendo um objeto estruturado de interesse (nesse caso, o avião), consideramos um particionamento S dessa imagem e uma rotulação f para as regiões produzidas pela segmentação. Em seguida, calculamos as medidas referentes às variáveis aleatórias descritas pelo GER segundo os dados de S e f, para cada conjunto de regiões de mesmo rótulo, armazenando-os como atributos de nós e arcos no respectivo grafo de observação.

consequentemente, os respectivos grafos de observação conterão medidas diferentes para cada objeto (nó) e suas relações com os demais (arcos).

Essa representação pressupõe uma rotulação f da imagem de entrada P, que é interpretada como uma possível observação do modelo estatístico-relacional. Essa rotulação pode ser obtida inicialmente de modo aleatório, ou segundo alguma outra estratégia de inicialização.

Portanto, a partir de P, S, e f, podemos definir um *grafo de observação* como sendo uma quádrupla $g_{obs} = (V_{obs}, E_{obs}, \vec{x}, \vec{y})$, em que:

- V_{obs} é o conjunto de nós de g_{obs}. O número total de nós é denotado por $|V_{obs}|$ e $|V_{obs}| = |L^*|$. A notação $v^i \in V_{obs}$, $i = 1, \ldots, |V_{obs}|$ será adotada para identificar um nó do conjunto. Cada nó v^i representa um elemento primitivo l e suas respectivas regiões expandidas s_l de S;

- E_{obs} é o conjunto de arcos de g_{obs}. Um arco é criado entre os nós representantes de dois conjuntos de regiões expandidas, referentes a um par de elementos primitivos que estão relacionados em G_m. O número total de arcos é denotado por $|E_{obs}|$ e $|E_{obs}| = |E_m|$. A notação $e^{ij} = (v^i, v^j)$ será utilizada para identificar um arco cujas extremidades são os nós $v^i \in V_{obs}$ (ponta

inicial) e $v^j \in V_{obs}$ (ponta final). A notação e^j, $j = 1, \ldots, |E_{obs}|$, também será aplicada quando o contexto não precisar os nós que compõem o arco;

- $\vec{x}(v^i) = [x_1(v^i), \ldots, x_{v_{V_m}}(v^i)]$ é uma função que devolve o vetor de valores observados x_p, $p = 1, \ldots, v_{V_m}$, relativos às variáveis aleatórias X_i que descrevem as propriedades dos elementos primitivos do modelo. Esses valores são calculados para o conjunto de regiões expandidas correspondentes ao nó v^i;

- $\vec{y}(e^{ij}) = [y_1(e^{ij}), \ldots, y_{v_{E_m}}(e^{ij})]$ é uma função que devolve o vetor de valores observados y_q, $q = 1, \ldots, v_{E_m}$, relativos às variáveis aleatórias Y_j que descrevem as relações entre os elementos primitivos do modelo. Esses valores são calculados para o par de conjuntos de regiões expandidas representadas pelos nós conectados por e^{ij}.

CRIAÇÃO DO GRAFO DE OBSERVAÇÃO. Como mencionado anteriormente, um grafo de observação exprime uma segmentação S associada a uma rotulação possível f, cujo domínio é L^*, de uma imagem de entrada P. Seja P' uma imagem contendo as regiões produzidas pela partição S, rotuladas segundo f. Então o grafo de observação extraído de P e P', segundo a estrutura de G_m dada por sua matriz de adjacências mtz_adj, segue os passos do algoritmo apresentado na Figura 8. Observe que, para cada rotulação possível, obtém-se um grafo com atributos diferentes.

O algoritmo cria cada nó do grafo segundo os elementos primitivos modelados e o insere na estrutura de dados. Em seguida, ele calcula as observações referentes a uma dada variável aleatória X_{idx} referente a um elemento primitivo l_i representado pelo modelo (linhas 2-7). Esse cálculo é feito sobre as regiões de P' que tenham recebido o rótulo associado a l_i, e segundo os respectivos dados originais da imagem original P (ex.: intensidade). O procedimento para criar os arcos é similar (linhas 9-17), e o cálculo das observações referentes às relações depende das regiões rotuladas como l_i e l_j em P', segundo os dados de P. Como medimos observações, os procedimentos COMPX e COMPY são os mesmos utilizados na etapa de aprendizagem supervisionada da criação do GER G_m.

3.3.3 *Formulação probabilística*

Na abordagem estatística, o reconhecimento de objetos se dá através do cálculo de observações das características descritas pelo modelo, e conseguinte classificação dessas observações segundo algum classificador probabilístico. Por outro lado, na abordagem estrutural, esse reconhecimento se dá através de casamento entre grafos (Capítulo 2), dado que o modelo e a imagem são representados através de grafos.

```
criaGrafoObservacao(P, P', L*, mtz_adj)
// Construção de g_obs
for i = 1 to |L*|
    v^i = Nó(l_i)
    V_obs = V_obs ∪ {v^i}
    // para cada observação x_idx associada a l_i em P'
    for idx = 1 to ν_{V_m}
        x_idx(v^i) = compX(l_i, idx, P[k], P'[k])
// para cada par de objetos relacionados segundo mtz_adj
for i = 1 to |L*|
    for j = 1 to |L*|
        if mtz_adj[i][j] == true
            e^ij = Arco(l_i, l_j)
            E_obs = E_obs ∪ {e^ij}
            // para cada observação y_idx associada ao par
            // de elementos l_i, l_j em P'
            for idx = 1 to ν_{E_m}
                y_idx(e^ij) = compY(l_i, l_j, idx, P[k], P'[k])
return g_obs
```

Figura 8: Criação de um GRA de observação (g_{obs}).

No conceito estatístico-estrutural, temos um modelo que combina a representação por grafo às estatísticas de características de aparência e relacionais do objeto. Portanto, dado um grafo de observação para uma imagem de entrada P, podemos formular o reconhecimento baseado em GER como um casamento entre grafos num arcabouço probabilístico, com restrições estruturais.

Logo, queremos avaliar a compatibilidade estatística entre um dado g_{obs} e o modelo G_m. Para isso, interpretamos g_{obs} como sendo uma potencial *observação* do GER modelo.

Para determinar a validade dessa hipótese, desejamos avaliar a probabilidade de que o grafo de observação g_{obs} tenha sido gerado pela descrição estatística contida no grafo modelo G_m. Nesse contexto probabilístico, definimos a probabilidade dada pela Equação 3.3 para medir a compatibilidade entre modelo e observação:

$$\begin{aligned} P(G_m = g_{obs}) \quad = \quad & P(\overrightarrow{X}(v_m^1) = \overrightarrow{x}(v^1), \ldots, \overrightarrow{X}(v_m^{|V_m|}) = \overrightarrow{x}(v^{|V_m|})), \\ & \overrightarrow{Y}(e_m^1) = \overrightarrow{y}(e^1), \ldots, P(\overrightarrow{Y}(e_m^{|E_m|}) = \overrightarrow{y}(e^{|E_m|})) \end{aligned} \tag{3.3}$$

Essa probabilidade conjunta é a fórmula mais genérica relacionando todas as variáveis que compõem os grafos modelo e de observação.

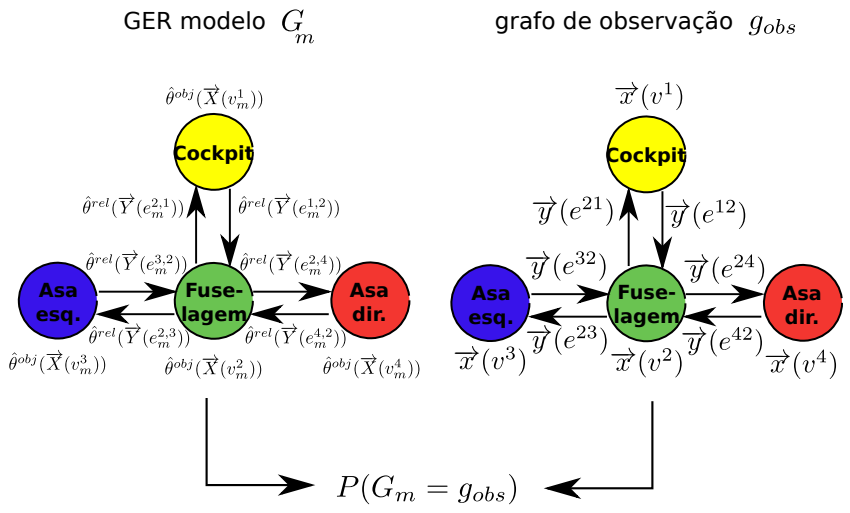


Figura 9: Ilustração do cálculo da probabilidade de um dado grafo de observação g_{obs} ter sido gerado pelo modelo expresso por G_m.

Inicialmente, podemos assumir que todas as variáveis dessa formulação são mutuamente dependentes umas das outras, ou seja, existe uma dependência total entre os elementos primitivos do modelo. Nesse caso, para calcular tal probabilidade, é possível adotar uma distribuição multivariada, como uma Normal ou algum modelo misto, estimar seus parâmetros, e aplicar a fórmula diretamente às medidas de entrada disponíveis. A vantagem dessa abordagem é a facilidade no cômputo do valor de $P(G_m = g_{obs})$ para cada observação, uma vez que os parâmetros da distribuição tenham sido estimados.

Contudo, como a dimensionalidade associada aos vetores aleatórios de G_m e de g_{obs} implicam numerosas variáveis, estimar os parâmetros se torna um problema matemática e computacionalmente intenso e difícil, exigindo também a disponibilidade de um grande número de amostras para que sejam estimados corretamente.

Portanto, assim como nos tópicos acerca de grafos aleatórios, uma questão naturalmente evocada é como expressar a probabilidade $P(G_m = g_{obs})$ de modo que esta faça uso de toda a informação possível transmitida pelo modelo e que flexibilize a condição de total dependência entre as variáveis envolvidas.

Há diversas maneiras de se calcular a probabilidade desejada sem relacionar todas as variáveis simultaneamente, buscando um equilíbrio entre representação e custo computacional. Logo, é possível escrever $P(G_m = g_{obs})$ de formas distintas, o que também pode ser entendido como um modo de projetar vários tipos de *funções-objetivos*.

Esse fato permite derivar fórmulas que atendem requisitos desejáveis previamente especificados, tais como convergência (i.e., que boas segmentações correspondam a altas probabilidades), robustez (pro-

babilidades que reflitam uma tolerância a variações de iluminação, transformações afins na imagem, etc.), bom desempenho computacional, projeto algorítmico facilitado (por exemplo, fácil paralelização), dentre outros.

A seguir, analisaremos duas hipóteses para amenizar a condição de dependência total.

3.3.4 *Hipótese de independência mútua*

No outro extremo da dependência total está a hipótese de que todas os elementos primitivos do modelo são independentes uns dos outros, e de que o conhecimento acerca de um deles nada revela sobre os demais. Essa simplificação conceitual é expressa na fórmula de $P(G_m = g_{obs})$ através da condição de independência mútua entre os elementos primitivos modelados, resultando na seguinte equação:

$$\begin{aligned} P(G_m = g_{obs}) \quad = \quad & \prod_{i=1,\ldots,|V_m|} P(\overrightarrow{X}(v_m^i) = \overrightarrow{x}(v^i)) \\ & \prod_{j=1,\ldots,|E_m|} P(\overrightarrow{Y}(e_m^j) = \overrightarrow{y}(e^j)) \end{aligned} \tag{3.4}$$

Em particular, o interesse nessa simplificação é sua facilidade de cálculo, já que todos os termos correspondem a probabilidades independentes, cada uma proveniente de um único elemento primitivo. Contudo, por esse mesmo fato, certas dependências inerentes aos elementos primitivos podem não ser explicitamente consideradas na modelagem probabilística.

Apesar dessa aparente restrição dessa condição, as informações estruturais e relacionais entre os elementos primitivos são expressas através das probabilidades das variáveis associadas aos arcos, e não através de dependências estatísticas entre os elementos primitivos.

Note que podemos supor ainda que as variáveis aleatórias referentes à características de um elemento primitivo ou relação são também mutuamente independentes. Portanto, cada termo da Equação 3.4 é escrito como:

$$P(\overrightarrow{X}(v_m^i) = \overrightarrow{x}(v^i)) \quad = \quad \prod_{p=1,\ldots,\nu_{V_m}} P(X_p(v_m^i) = x_p(v^i)), \tag{3.5}$$

ou

$$P(\overrightarrow{Y}(e_m^{ij}) = \overrightarrow{y}(e^{ij})) \quad = \quad \prod_{q=1,\ldots,\nu_{E_m}} P(Y_q(e_m^{ij}) = y_q(e^{ij})). \tag{3.6}$$

Interpretando a fórmula de maneira prática, o fato de considerarmos os vetores aleatórios de aparência e relacionais de um elemento primitivo independentemente dos demais pode ser compreendido,

por exemplo, como uma indicação de que o conhecimento da proporção entre a asa direita e a fuselagem não fornece qualquer pista sobre a proporção entre a fuselagem e o *cockpit*.

OTIMIZAÇÃO. Os problemas de casamento entre grafos exigem a definição de uma medida de similaridade para poder compará-los, e a definição de um mapeamento adequado entre ambos deve ser realizada por algum algoritmo de otimização adequado. Da mesma forma, nesta proposta probabilística, a probabilidade $P(G_m = g_{obs})$ deve ser maximizada através de algum algoritmo de otimização. Na literatura, encontramos trabalhos comparativos desses algoritmos [CJBBL05], e a escolha do mais adequado poderá depender da aplicação de interesse.

3.3.5 *Hipótese de independência condicional*

A suposição de independência mútua entre os elementos primitivos, embora válida, não representa necessariamente todas as interações existentes entre os elementos representados pelo modelo. Devido à estrutura do modelo e às relações partilhadas entre os elementos primitivos, podemos assumir a hipótese de independência condicional entre os elementos primitivos e utilizar esse conhecimento no processo de reconhecimento. Para explorar essa hipótese, vamos recorrer à teoria de campos aleatórios de Markov (MRF, de *Markov random field*) [Per98, Li09], ou Markovianos. A seguir, apresentaremos um estudo introdutório acerca de como desenvolver a hipótese de independência condicional segundo essa teoria.

Campos aleatórios Markovianos

Considere o conjunto de regiões refinadas resultantes de uma partição S de P. Considere também um grafo de observação g_{obs} derivado de S, mas não associado a uma rotulação. Nesse caso $|V_{obs}| = |S|$, e a sua respectiva topologia deriva da adjacência entre as regiões de S. As observações das variáveis aleatórias do modelo, associadas aos nós e arcos de g_{obs}, também são calculadas a partir das regiões refinadas de S.

Seja ainda o sistema de vizinhança derivado da topologia do grafo g_{obs}:

$$N_{V_{obs}} = \{N_{v^i} : v^i \in V_{obs}\}, \tag{3.7}$$

em que,

$$N_{v^i} = \{v^j : v^i, v^j \in V_{obs}, v^i \neq v^j \text{ e } \exists e^{ij} \in E_{obs} \text{ e } \exists e^{ji} \in E_{obs}\}. \tag{3.8}$$

Convém observar que V_{obs} é finito e N_{v^i} é limitado.

A cada v^i, vamos associar uma variável aleatória F_i, cujas realizações f_i assumem valores em L^*, discreto. Então, definimos o campo aleatório Markoviano $F = (F_1, \ldots, F_{|V_{obs}|})$, cujas realizações $f = (f_1, \ldots, f_{|V_{obs}|})$ assumem valores em $\Omega = |L^*|^{|V_{obs}|}$, e cujo sistema de vizinhança é dado por $N_{V_{obs}}$.

Portanto, um campo aleatório de Markov consiste num modelo gráfico representado por um grafo não-dirigido $G = (V, E)$ em que V é um conjunto de nós, que aqui representam variáveis aleatórias associadas aos rótulos das regiões refinadas, e E é um conjunto de arestas, que representam as dependências condicionais entre os rótulos das regiões, dadas por $N_{V_{obs}}$.

A realização f é chamada de *configuração* de F, dentre todas as configurações possíveis, que aqui será denotada por F^{all}. Esta suposição é possível porque podemos verificar as seguintes propriedades:

- $P(F = f) > 0, \forall f \in F^{all}$: supomos que todas as configurações do campo aleatório são prováveis, embora algumas delas sejam menos desejáveis do que outras;
- $P(F_i = f_i | F_j = f_j, v^i \neq v^j) = P(F_i = f_i | F_j = f_j, v^j \in N(v^i)), \forall v^i \in V_{obs}$ e $f \in F^{all}$: o rótulo atribuído a um sítio do campo aleatório depende apenas dos rótulos atribuídos a seus vizinhos diretos.

Graças ao teorema de Hammersley-Clifford, sabemos que uma configuração de um campo aleatório Markoviano obedece a uma distribuição de Gibbs:

$$P(F = f) = \frac{1}{Z} \exp(-U(f)/T), \tag{3.9}$$

em que Z e T são constantes. O termo $U(f)$ é uma função de energia equivalente à soma de todos os potenciais de cliques $\phi_c(f)$ presentes na configuração do campo Markoviano, i.e., uma fatoração do conjunto total de variáveis, como descrito na equação 3.10:

$$U(f) = \sum_{c \in C} \phi_c(f) \ , \tag{3.10}$$

em que C é o conjunto de cliques associados ao sistema de vizinhança do campo Markoviano, enquanto $\phi_c(f)$ é um potencial de clique, equivalente a medidas das interações locais entre os descritores das regiões refinadas e as de seus vizinhos.

Os potenciais aqui adotados são de ordem 1, $\phi_{c_1}(f_i)$, ou de ordem 2, $\phi_{c_2}(f_i, f_{i'})$, e, portanto, o conjunto de cliques considerados globalmente é dado por:

$$U(f) = \sum_{i \in C_1} \phi_{c_1}(f_i) + \sum_{\{i,i'\} \in C_2} \phi_{c_2}(f_i, f_{i'}). \tag{3.11}$$

Apenas esses dois tipos de cliques serão considerados, para simplificar a modelagem do problema. Como estamos lidando com relações, cliques de ordem maior do que 2 implicariam no tratamento de relações mais complexas do que relações binárias, e exigiriam adotar a estrutura de hipergrafos tanto para representar o GER, quanto o grafo de observação.

O potencial de ordem 1 (potencial simples) pode avaliar o conhecimento a priori acerca da aparência de um elemento primitivo, segundo observações de medidas provenientes diretamente da imagem P, como é o caso da intensidade. Por outro lado, através do potencial de ordem 2, podemos incentivar a rotulação uniforme de regiões refinadas vizinhas que pertençam a um mesmo elemento primitivo, de acordo com alguma medida relacional que seja aplicável entre regiões refinadas, tal como o gradiente entre as duas regiões.

Há um número exponencial de configurações possíveis para um dado campo Markoviano, o que equivale ao número de rotulações possíveis para as regiões de S. Logo, procuramos uma configuração ótima f^* de rótulos, no sentido de máximo *a posteriori*. Dentro do arcabouço Bayesiano, essa configuração é traduzida pela seguinte probabilidade:

$$f^* = \arg\max_{f \in F^{all}} P(F = f | V_{obs}, V_m) \tag{3.12}$$

e, aplicando-se a regra de Bayes, temos:

$$P(F = f | V_{obs}, V_m) \propto P(V_{obs} | F = f, V_m) P(f) \tag{3.13}$$

Seja $P(V_m = V_{obs} | F = f)$ a probabilidade de ocorrência de uma dada realização do campo aleatório, que, neste caso, corresponde a uma rotulação da imagem, dada a partição S. Assumimos que os atributos de cada região são independentes daqueles provenientes de outras regiões e escrevemos:

$$\begin{aligned} P(V_{obs} | F = f, V_m) &= \prod_{i=1}^{|V_{obs}|} P(v^i | f_i) \\ &= \prod_{i=1}^{|V_{obs}|} \prod_{k=1}^{\nu} P(x_k(v^i) | f(v^i)) \end{aligned} \tag{3.14}$$

Cada probabilidade condicional dada por $P(x_k(v^i) | f(v^i), X_k(v_m))$ pode ser facilmente obtida a partir das informações estatísticas armazenadas em G_m para cada órgão. Como os parâmetros $\hat{\theta}^{obj}(\overrightarrow{X}(v_m))$ são estimados para cada variável aleatória, e temos disponíveis os respectivos valores observados $\overrightarrow{x}(v^i)$, podemos calcular $P(\overrightarrow{X}(v_m^i) = \overrightarrow{x}(v^i))$.

Através das equações 3.9 e 3.14, podemos resolver a equação 3.12 através de um método de otimização para campos Markovianos, tais como o tradicional recozimento simulado [DHS00, Li09] (*simulated*

annealing), alfa-expansões, ou trocas alfa-beta (*alpha-beta swaps*), propostas por Boykov, Veksler e Zabih [BVZ01].

Este é um problema clássico de segmentação de imagens através do arcabouço MAP-MRF, de *Maximum a posteriori-Markov Random Field*. Contudo, apenas alguns aspectos do conhecimento a priori existente são utilizados. Como a topologia do campo Markoviano é dada, nesse caso, pelas adjacências entre regiões refinadas de uma partição inicial S de P, o algoritmo de otimização para o cálculo da Equação 3.9 não se beneficiará diretamente das características relacionais. Isso ocorre porque as relações calculadas entre regiões refinadas (micro-regiões) adjacentes não são diretamente comparáveis àquelas que caracterizam os pares de elementos primitivos (macro-regiões), como discutimos na Seção 3.3.1.

Portanto, o desafio nesse formalismo, sob a hipótese de independência condicional, consiste em incorporar gradativamente as informações relacionais de acordo com as regiões expandidas que se formam a cada passo do algoritmo de otimização, permitindo que essas informações o auxiliem a percorrer o espaço de configurações de modo eficaz e coerente com as relações do modelo.

3.4 RESULTADOS EXPERIMENTAIS

Para testar experimentalmente a hipótese de independência dentro da interpretação probabilística para o reconhecimento do conteúdo de uma imagem de entrada em função do GER modelo, realizamos um conjunto de simulações com dados sintéticos no ambiente de programação *R*. O objetivo desses experimentos consistiu em avaliar o comportamento da probabilidade $P(G_m = g_{obs})$ segundo várias configurações dos dados e suas respectivas segmentações, e verificar sua plausibilidade como uma função-objetivo.

3.4.1 *Dados simulados*

Os dados simulados representam um objeto estruturado tridimensional composto de quatro elementos primitivos l_1, l_2, l_3, l_4. Cada elemento primitivo está associado às seguintes características de aparência:

- intensidade, representada pela variável aleatória X_1;
- coordenadas 3D (x,y,z) do centróide, representadas pelas variáveis aleatórias X_2, X_3, X_4, respectivamente.

Relações entre pares de elementos primitivos estão representadas pelas seguintes características quantitativas:

- distância entre centróides, representada pela variável aleatória Y_1;

- ângulo entre o plano xy e a reta suporte que conecta os centróides de cada elemento primitivo, representado pela variável aleatória Y_2.

O GER modelo G_m que representa a descrição conceitual do modelo simulado corresponde ao grafo ilustrado pela Figura 10. Esse grafo é definido por:

- $V_m = \{v_m^1, v_m^2, v_m^3, v_m^4\}$, vértices representando os quatro elementos primitivos l_1, l_2, l_3 e l_4 respectivamente, ;
- $E_m = \{(v_m^1, v_m^2), (v_m^1, v_m^4), (v_m^3, v_m^4)\}$, arestas representando pares de elementos primitivos relacionados;
- $\hat{\theta}^{obj}(\overrightarrow{X}(v_m)) = [\hat{\theta}(X_1(v_m)), \ldots, \hat{\theta}(X_4(v_m))]$, $v_m \in V_m$, em que cada $\overrightarrow{X}_i$, $i = 1 \ldots 4$, é o vetor de parâmetros estimados para cada distribuição Normal associada aos atributos de aparência do elemento primitivo representado por v_m;
- $\hat{\theta}^{rel}(\overrightarrow{Y}(e_m)) = [\hat{\theta}(Y_1(e_m)), \hat{\theta}(Y_2(e_m))]$, $e_m \in E_m$, em que cada $\overrightarrow{Y}_j$, $j = 1, 2$, é o vetor de parâmetros estimados para cada distribuição Normal associada aos atributos relacionais entre elementos primitivos representados pelos nós conectados por e_m. A magnitude e o ângulo médios para cada relação foram calculados com base nas coordenadas dos centróides dos elementos primitivos em questão. Já a variância de cada relação foi estipulada manualmente para cada simulação.

Cada GRA criado para simular o grafo obtido para uma dada segmentação (ou seja, uma observação), consistiu num grafo g_{obs} tal que:

- $V_{obs} = \{v^1, v^2, v^3, v^4\}$, representando quatro conjuntos de regiões expandidas referentes aos quatro elementos primitivos modelados;
- $E_{obs} = \{(v^1, v^2), (v^1, v^4), (v^3, v^4)\}$, relações bidirecionais entre os mesmos pares de elementos primitivos relacionados no modelo;
- $\overrightarrow{x}(v^i) = [x_1(v^i), \ldots, x_4(v^i)]$, em que cada $x_k(v^i)$, $k = 1, \ldots, 4$ é uma amostra sorteada a partir da distribuição Normal dada por $X_i(v_m^i)$;
- $\overrightarrow{y}(e^j) = [y_1(e^j), y_2(e^j)]$, em que cada medida é calculada segundo as coordenadas dos centróides dos elementos primitivos em questão.

Um exemplo ilustrativo para g_{obs} é dado também na Figura 10. Nesse exemplo, o vértice v^1 compreende três regiões refinadas, rotuladas como pertinentes ao elemento primitivo l_1; o vértice v^2 compreende duas regiões refinadas, rotuladas como pertinentes ao elemento

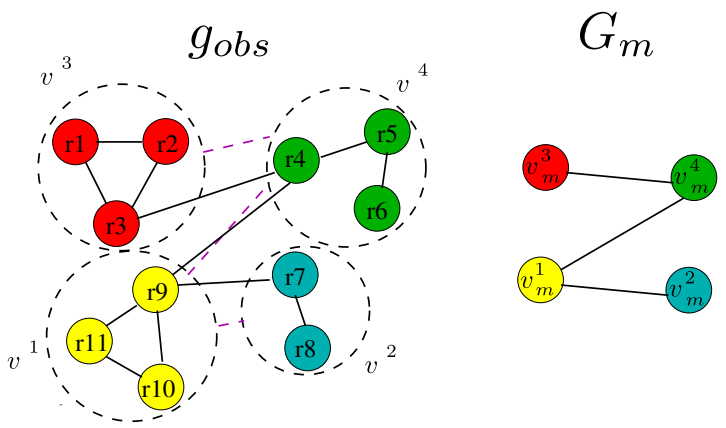


Figura 10: Representação gráfica dos grafos simulados: um exemplo de GRA g_{obs} (à dir.), em que cada nó corresponde a uma amostra obtida a partir de um elemento primitivo representado pelo GER modelo G_m (à esq.). As cores dos nós estabelecem uma relação entre os elementos primitivos representados no modelo e os rótulos associados a cada amostra.

primitivo l_2, e assim por diante. Nas simulações realizadas, cada região refinada corresponde a uma amostra gerada para um dado elemento primitivo.

Para cada experimento, assumimos distribuições Normais para as variáveis aleatórias que modelam os atributos de aparência e relacionais do GER, e seus respectivos parâmetros μ e σ foram fixados manualmente de acordo com o efeito que desejamos analisar.

3.4.2 *Configurações do experimento*

Cada experimento consistiu de duas etapas. A primeira delas foi responsável por gerar um conjunto de amostras para cada elemento estruturado modelado, às quais foram associados os respectivos rótulos verdadeiros. Em seguida, os atributos relativos às variáveis aleatórias de interesse foram calculados para cada conjunto de amostras geradas de mesmo rótulo. Finalmente, foi calculada $P(G_m = g_{obs})$, ou seja, a probabilidade da solução correta (*ground truth*). Assumimos a hipótese de independência mútua entre as variáveis do modelo, o que leva ao cálculo da seguinte probabilidade para cada observação:

$$P(G_m = g_{obs}) = \prod_{i=1,\ldots,4} P(\overrightarrow{X}(v_m^i) = \overrightarrow{x}(v^i)) \prod_{j=1,\ldots,3} P(\overrightarrow{Y}(e_m^j) = \overrightarrow{y}(e^j)) \tag{3.15}$$

Por sua vez, cada atributo foi também considerado independente dos demais. Assim, a probabilidade das medidas de atributos é também escrita como o produto de suas probabilidades individuais. No caso dos atributos de aparência de um elemento primitivo representado por v^i_m, temos:

$$P(\overrightarrow{X}(v^i_m) = \overrightarrow{x}(v^i)) = P(X^i_1 = x^i_1)P(X^i_2 = x^i_2)P(X^i_3 = x^i_3)P(X^i_4 = x^i_4) \tag{3.16}$$

e, no caso dos atributos relacionais entre um par de elementos primitivos, temos:

$$P(\overrightarrow{Y}(e^j_m) = \overrightarrow{y}(e^j)) = P(Y^j_1 = y^j_1)P(Y^j_2 = y^j_2) \tag{3.17}$$

Em resumo, para cada elemento primitivo l_i, essa primeira etapa consiste em:

1. gerar n amostras para l_i, representado por seus atributos de aparência e relacionais e rotulá-los com o rótulo l_i;
2. calcular os valores médios de todos os seus atributos de acordo com todas as amostras geradas para esse elemento primitivo;
3. calcular as respectivas medidas dos atributos relacionais induzidos para essa amostragem, segundo as relações dadas pela topologia de G_m.

A segunda etapa de cada experimento consistiu na introdução de erros na rotulação associada a cada amostra, seguida da análise de como esses erros influenciam a probabilidade global de um dado grafo de observação. Desejamos testar a hipótese de que, conforme o número de erros aumenta, a probabilidade $P(G_m = g_{obs})$ deveria decrescer. Para verificar essa hipótese, realizamos a troca sistemática de rótulos associados às mesmas amostras geradas na etapa precedente. A troca dos rótulos é feita de maneira gradual e crescente, ou seja, a cada teste, mais erros são introduzidos na rotulação. Além disso, cada teste efetuado com troca de m rótulos, $m = 1 \ldots namostras$, associados a m amostras distintas, é repetido k vezes, para que se calcule uma média dos valores obtidos para $P(G_m = g_{obs})$ e reduzir a possibilidade de gerar erros aleatórios pouco representativos.

Essa segunda etapa pode ser resumida nos seguintes passos:

repetir k **vezes**

1. trocar m rótulos, $m = 1 \ldots namostras$, $namostras = n \times |V_m|$ (introduzir erros, variando desde um único rótulo incorreto até a troca de todos os rótulos);

2. criar o respectivo GRA g_{obs}:
   - para cada elemento primitivo, calcular os atributos médios para cada conjunto de amostras que levam o seu respectivo rótulo;
   - para cada par de elementos primitivos relacionados segundo a topologia de G_m, calcular os atributos relacionais induzidos pelas amostras do passo anterior;

3. calcular $P(G_m = g_{obs})$, de acordo com a equação 3.4;

4. atualizar a média de $P(G_m = g_{obs})$ para m erros aleatórios.

Os experimentos relatados a seguir mostram os gráficos de $P(G_m = g_{obs})$ em função do número de regiões representantes de um elemento primitivo, denotado por n, e o número de testes repetidos k. Note que a variação de n simula o número de regiões refinadas produzidas pelo algoritmo de particionamento adotado na metodologia.

Em cada experimento, também foi calculado o valor de $P(G_m = g^*_{obs})$, que corresponde à probabilidade de uma observação, dada por g^*_{obs}, cujos rótulos estão todos corretos (*ground truth*) e cujos atributos correspondem exatamente às médias calculadas para cada atributo de cada elemento primitivo ou relação contido em G_m. Esse valor permite estimar um limite superior para $P(G_m = g^*_{obs})$, que é a probabilidade de uma solução perfeita segundo o modelo. Note que essa probabilidade não é necessariamente igual a 1, já que é resultado de um produto de probabilidades. Nos gráficos, $P(G_m = g^*_{obs})$ é representada por um ponto vermelho.

Finalmente, os dados amostrados em cada experimento (conforme a descrição da Etapa 1) são exibidos num gráfico em que cada ponto corresponde a uma amostra sorteada para um dado elemento primitivo (ou nó de G_m), identificado por uma cor específica. Pontos representados por uma estrela correspondem à média das amostras sorteadas.

3.4.3 *Simulação 1*

Objetivo: testar o comportamento de $P(G_m = g_{obs})$ sob a hipótese de independência mútua entre os nós e as variáveis aleatórias que representam atributos e relações. As variáveis aleatórias que representam o nível de cinza e as coordenadas espaciais do conjunto de elementos primitivos são regidas, neste caso, por distribuições de probabilidade que não se sobrepõem, e cujos desvios-padrões são relativamente pequenos, para representar elementos primitivos com características de pouca variabilidade.

- $\hat{\theta}^{obj}(\overrightarrow{X}(v_m^1)) = [X_1(v_m^1) \sim \mathcal{N}(128, 10), X_2(v_m^1) \sim \mathcal{N}(0, 0.01),$ $X_3(v_m^1) \sim \mathcal{N}(5, 0.01), X_4(v_m^1) \sim \mathcal{N}(0, 0.01)]$;
- $\hat{\theta}^{obj}(\overrightarrow{X}(v_m^2)) = [X_1(v_m^2) \sim \mathcal{N}(220, 15), X_2(v_m^2) \sim \mathcal{N}(10, 0.01),$ $X_3(v_m^2) \sim \mathcal{N}(5, 0.01), X_4(v_m^2) \sim \mathcal{N}(0, 0.01)]$;
- $\hat{\theta}^{obj}(\overrightarrow{X}(v_m^3)) = [X_1(v_m^3) \sim \mathcal{N}(75, 30), X_2(v_m^3) \sim \mathcal{N}(0, 0.01),$ $X_3(v_m^3) \sim \mathcal{N}(5, 0.01), X_4(v_m^3) \sim \mathcal{N}(10, 0.01)]$;
- $\hat{\theta}^{obj}(\overrightarrow{X}(v_m^4)) = [X_1(v_m^4) \sim \mathcal{N}(10, 5), X_2(v_m^4) \sim \mathcal{N}(10, 0.01),$ $X_3(v_m^4) \sim \mathcal{N}(4, 0.01), X_4(v_m^4) \sim \mathcal{N}(10, 0.01)]$;

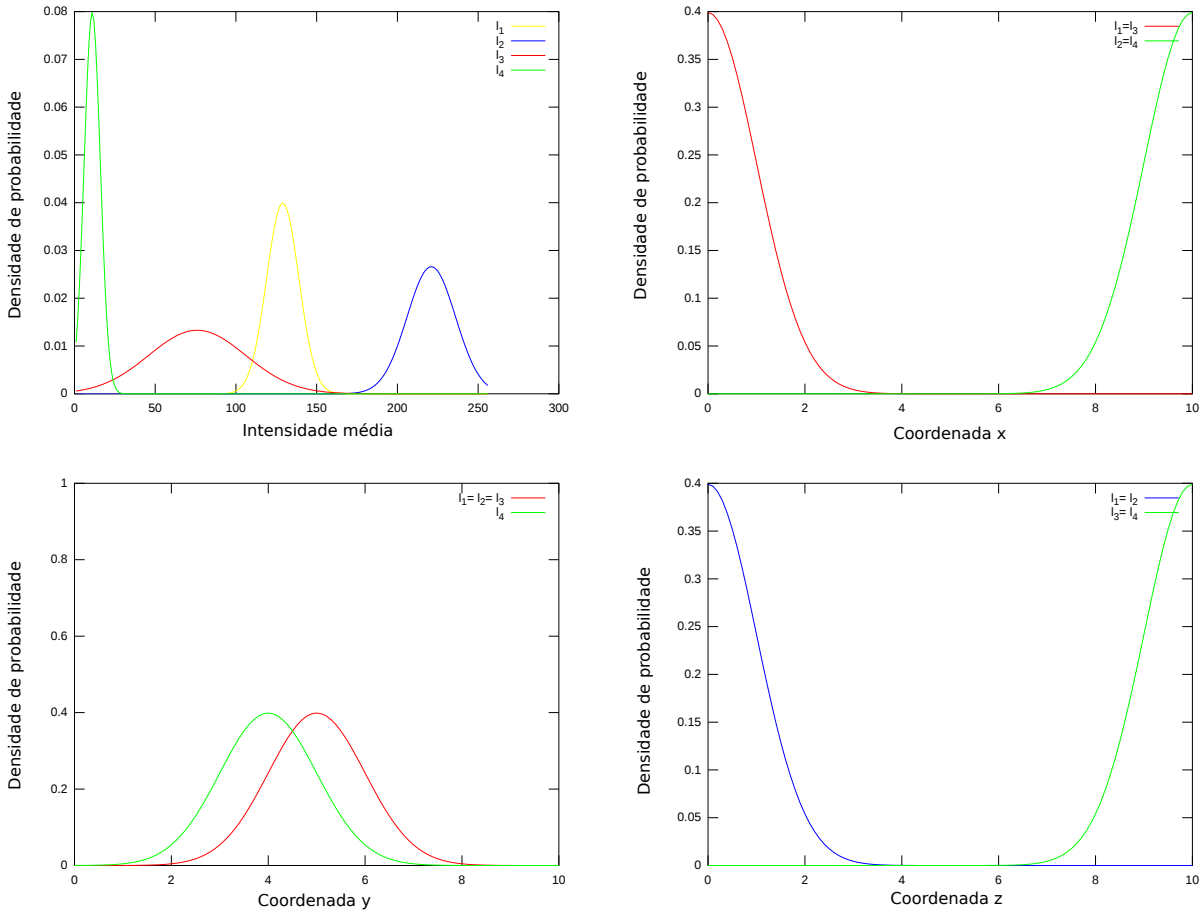


Figura 11: Simulação 1: densidades de probabilidade das variáveis aleatórias representantes das características de aparência de cada elemento primitivo modelado: l_1 (amarelo), l_2 (azul), l_3 (vermelho), l_4 (verde).

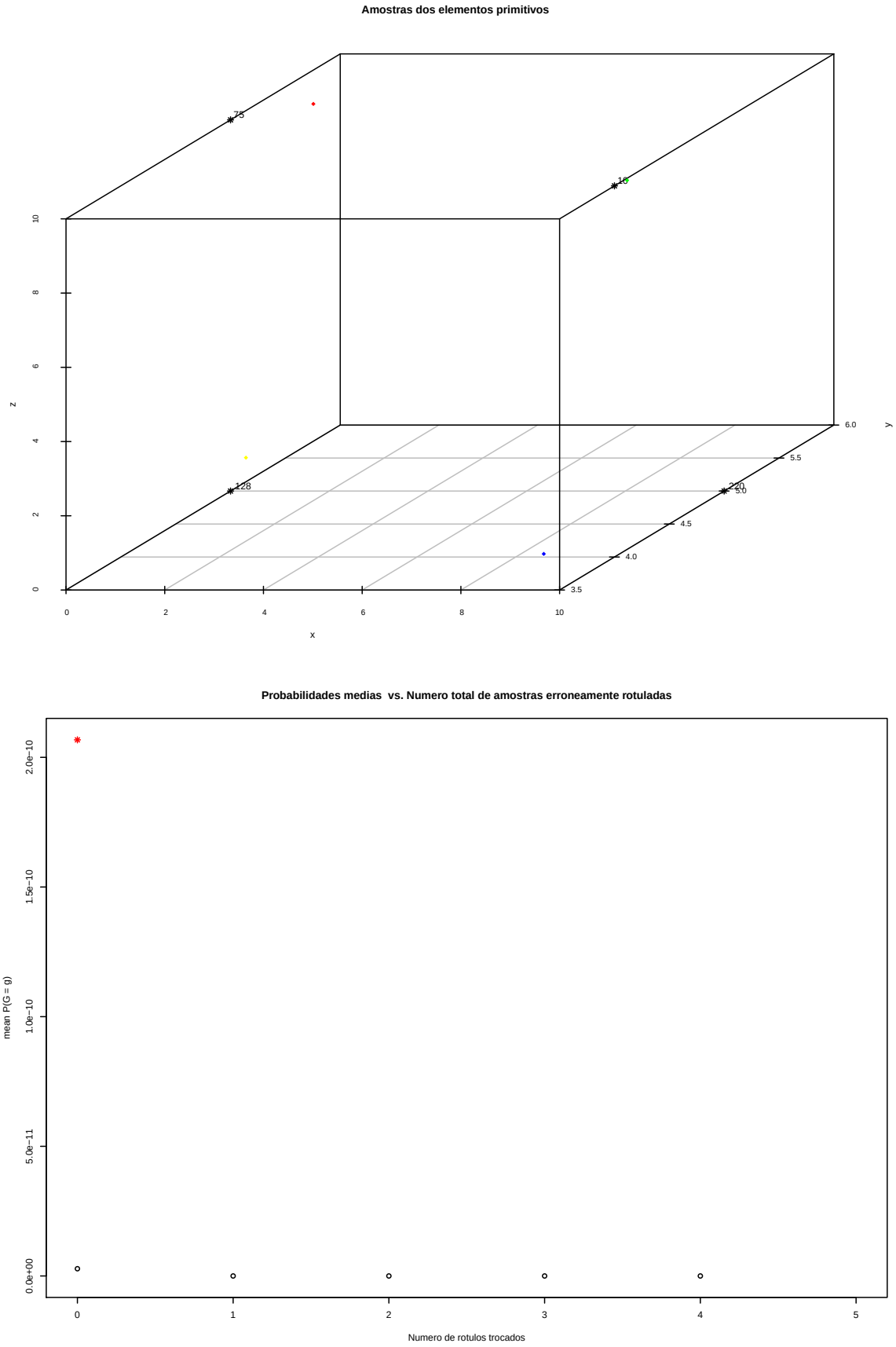


Figura 12: Resultados para Simulação 1: dados amostrados e respectivas curvas probabilidade $\times$ erros para $k = 1$ e $n = 1$.

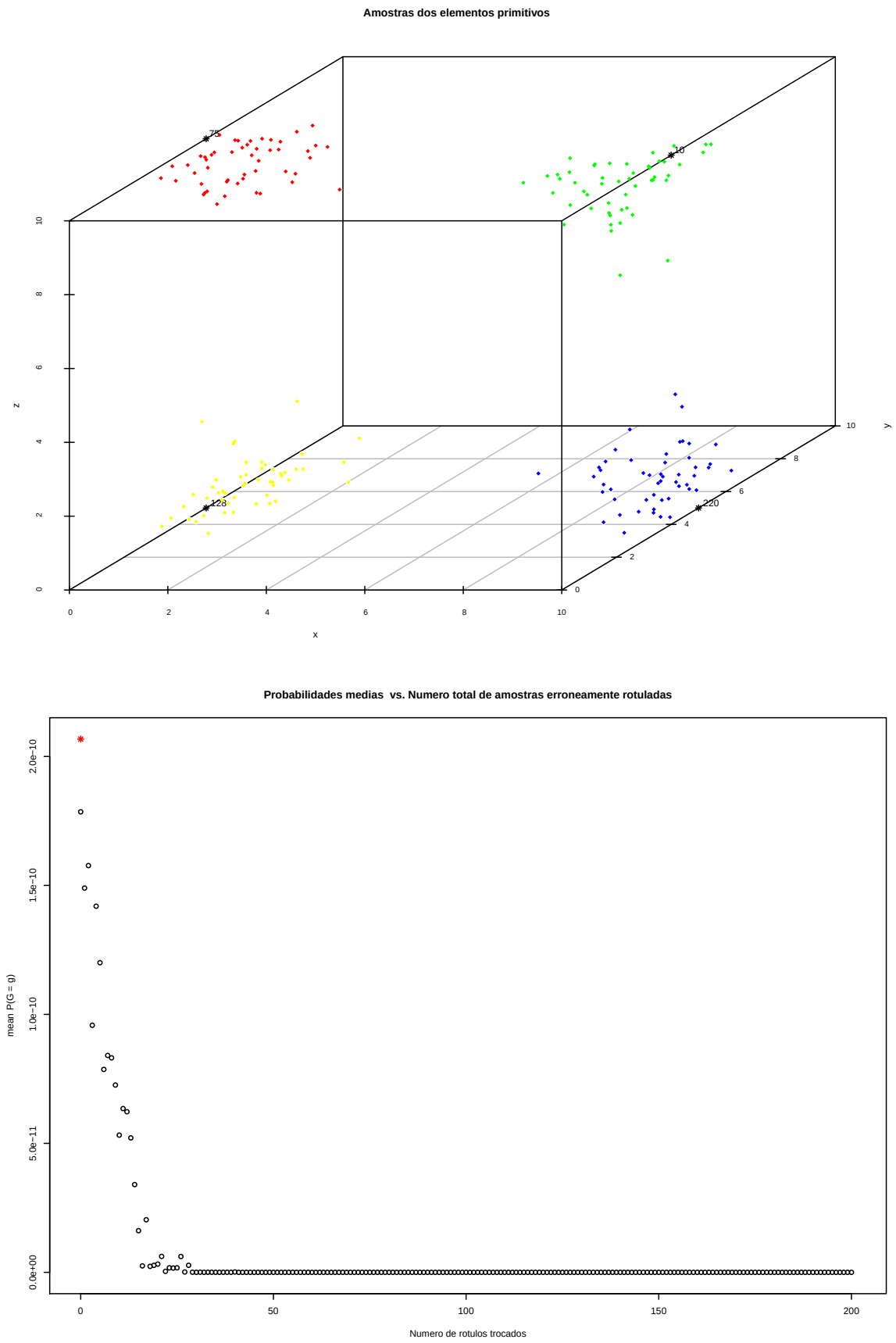


Figura 13: Resultados para Simulação 1: dados amostrados e respectivas curvas probabilidade $\times$ erros para $k = 5$ e $n = 50$.

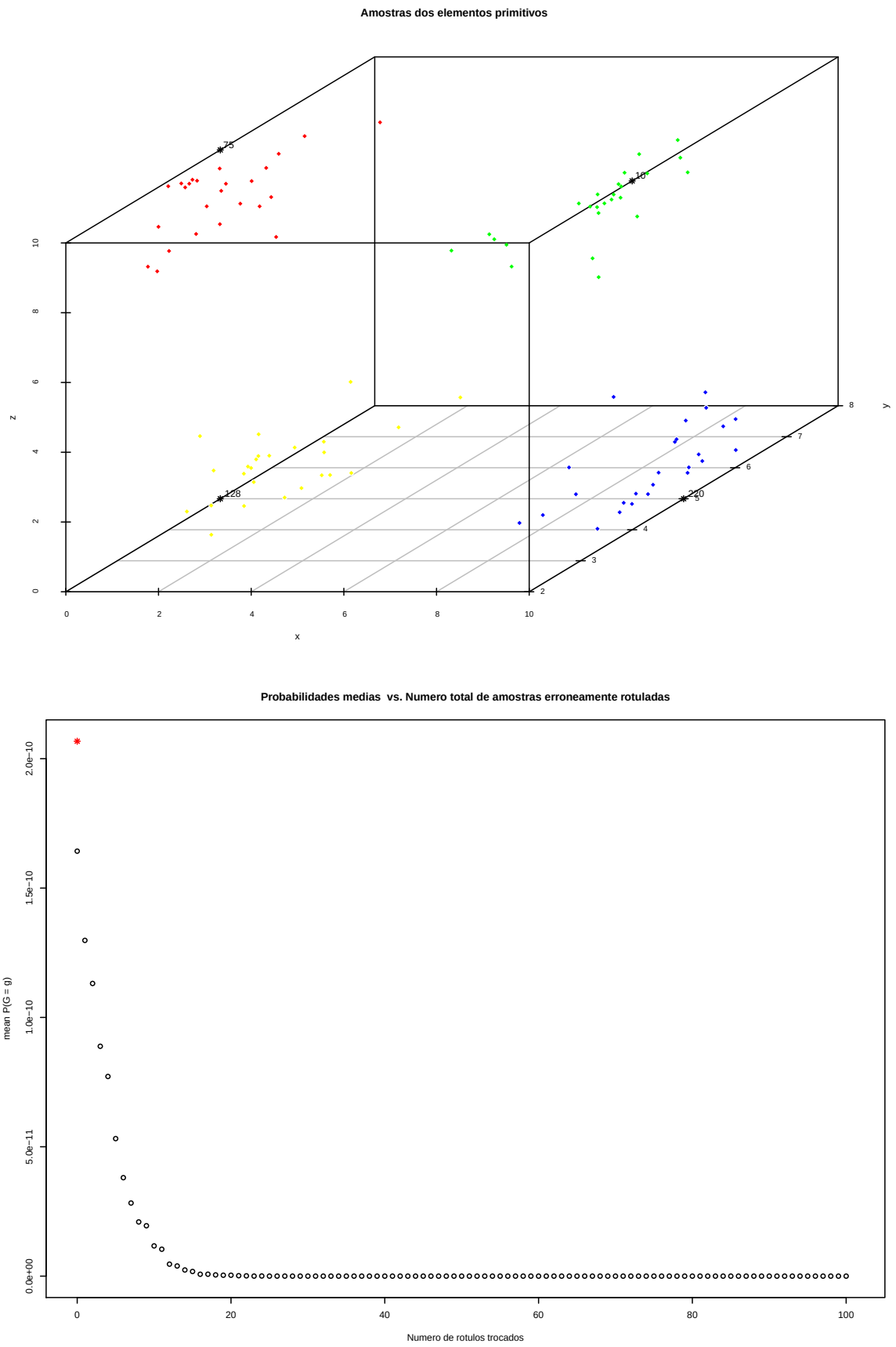


Figura 14: Resultados para Simulação 1: dados amostrados e respectivas curvas probabilidade $\times$ erros para $k = 100$ e $n = 25$.

3.4.4 *Simulação 2*

Objetivo: testar o comportamento de $P(G_m = g_{obs})$ sob a hipótese de independência mútua entre os nós e as variáveis aleatórias que representam atributos e relações. As variáveis aleatórias que representam os níveis de cinza dos elementos primitivos apresentam, neste caso, médias que não se assemelham, mas cujos desvios-padrões são amplos, para representar elementos primitivos com características de aparência distintas, mas de grande variabilidade. Por outro lado, as variáveis aleatórias que representam as coordenadas espaciais dos elementos primitivos são regidas por distribuições de probabilidade bastante distintas. Dessa forma, a posição dos elementos primitivos e suas respectivas relações deveriam compensar a similaridade de aparência dos mesmos, permitindo distinguir uma segmentação correta de outras menos desejáveis.

- $\hat{\theta}^{obj}(\overrightarrow{X}(v_m^1)) = [X_1(v_m^1) \sim \mathcal{N}(128, 50), X_2(v_m^1) \sim \mathcal{N}(0, 0.01),$ $X_3(v_m^1) \sim \mathcal{N}(5, 0.01), X_4(v_m^1) \sim \mathcal{N}(0, 0.01)]$;
- $\hat{\theta}^{obj}(\overrightarrow{X}(v_m^2)) = [X_1(v_m^2) \sim \mathcal{N}(220, 30), X_2(v_m^2) \sim \mathcal{N}(10, 0.01),$ $X_3(v_m^2) \sim \mathcal{N}(5, 0.01), X_4(v_m^2) \sim \mathcal{N}(0, 0.01)]$;
- $\hat{\theta}^{obj}(\overrightarrow{X}(v_m^3)) = [X_1(v_m^3) \sim \mathcal{N}(75, 40), X_2(v_m^3) \sim \mathcal{N}(0, 0.01),$ $X_3(v_m^3) \sim \mathcal{N}(5, 0.01), X_4(v_m^3) \sim \mathcal{N}(10, 0.01)]$;
- $\hat{\theta}^{obj}(\overrightarrow{X}(v_m^4)) = [X_1(v_m^4) \sim \mathcal{N}(10, 25), X_2(v_m^4) \sim \mathcal{N}(10, 0.01),$ $X_3(v_m^4) \sim \mathcal{N}(4, 0.01), X_4(v_m^4) \sim \mathcal{N}(10, 0.01)]$;

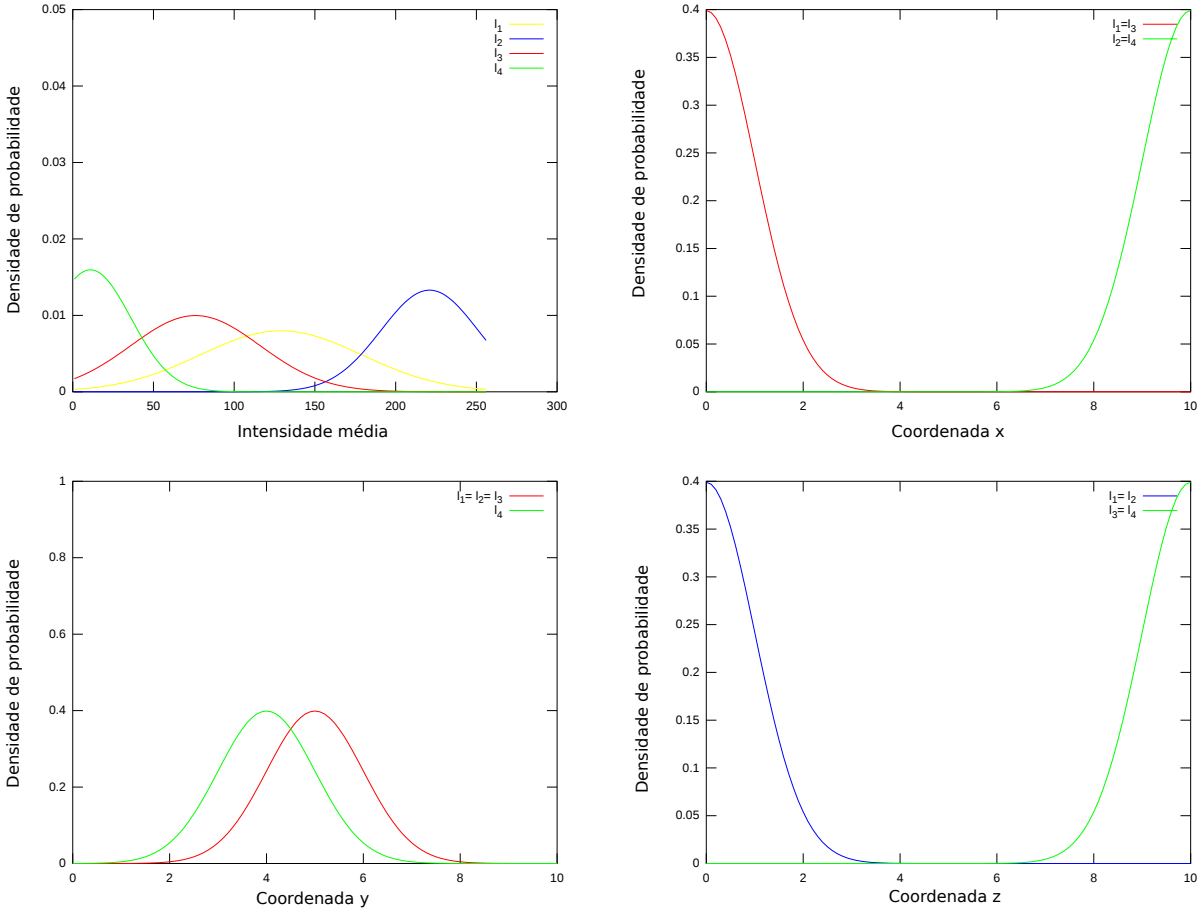


Figura 15: Simulação 2: densidades de probabilidade das variáveis aleatórias representantes das características de aparência de cada elemento primitivo modelado: l_1 (amarelo), l_2 (azul), l_3 (vermelho), l_4 (verde).

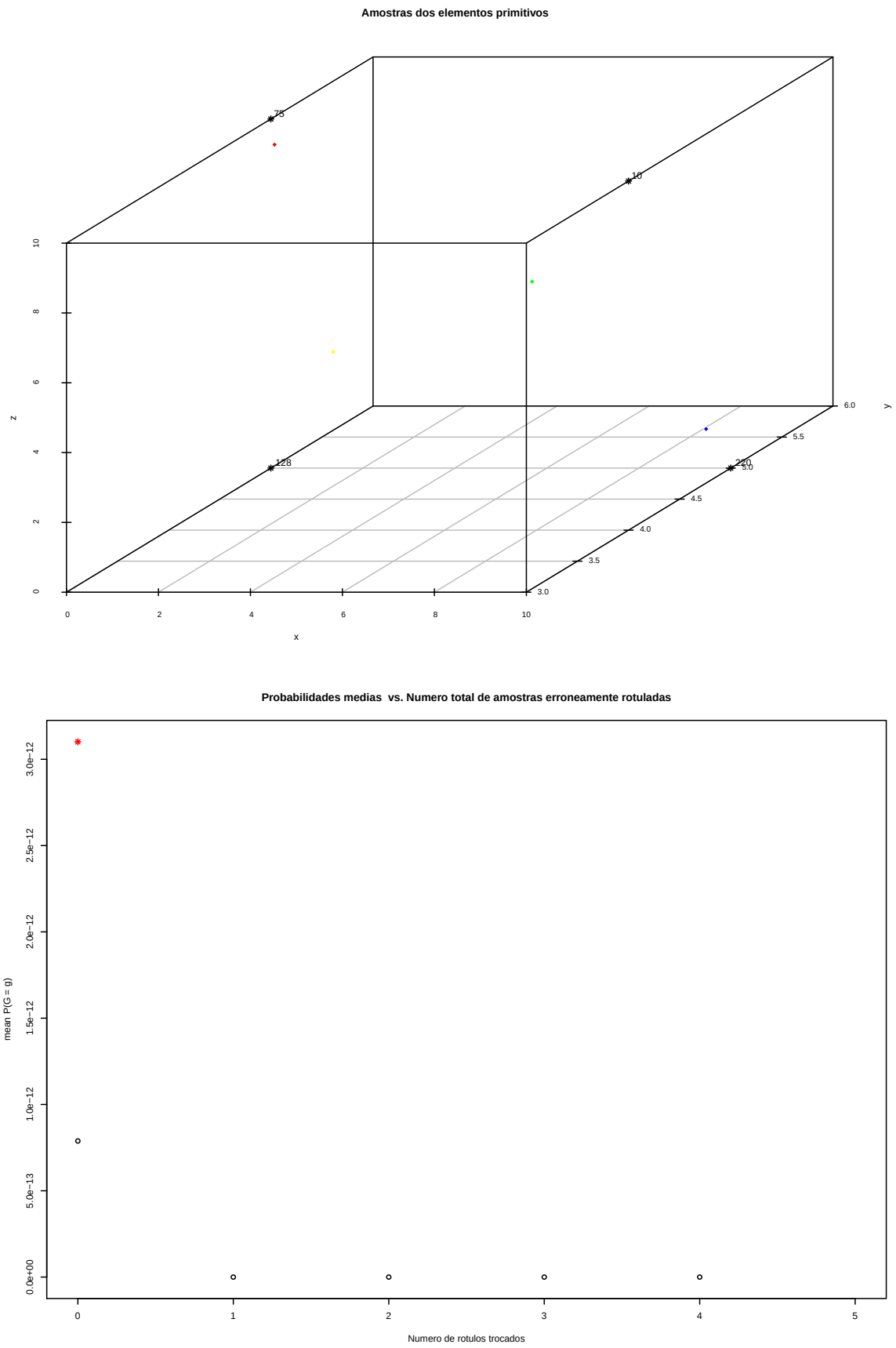


Figura 16: Resultados para Simulação 2: dados amostrados e respectivas curvas probabilidade $\times$ erros para $k = 1$ e $n = 1$.

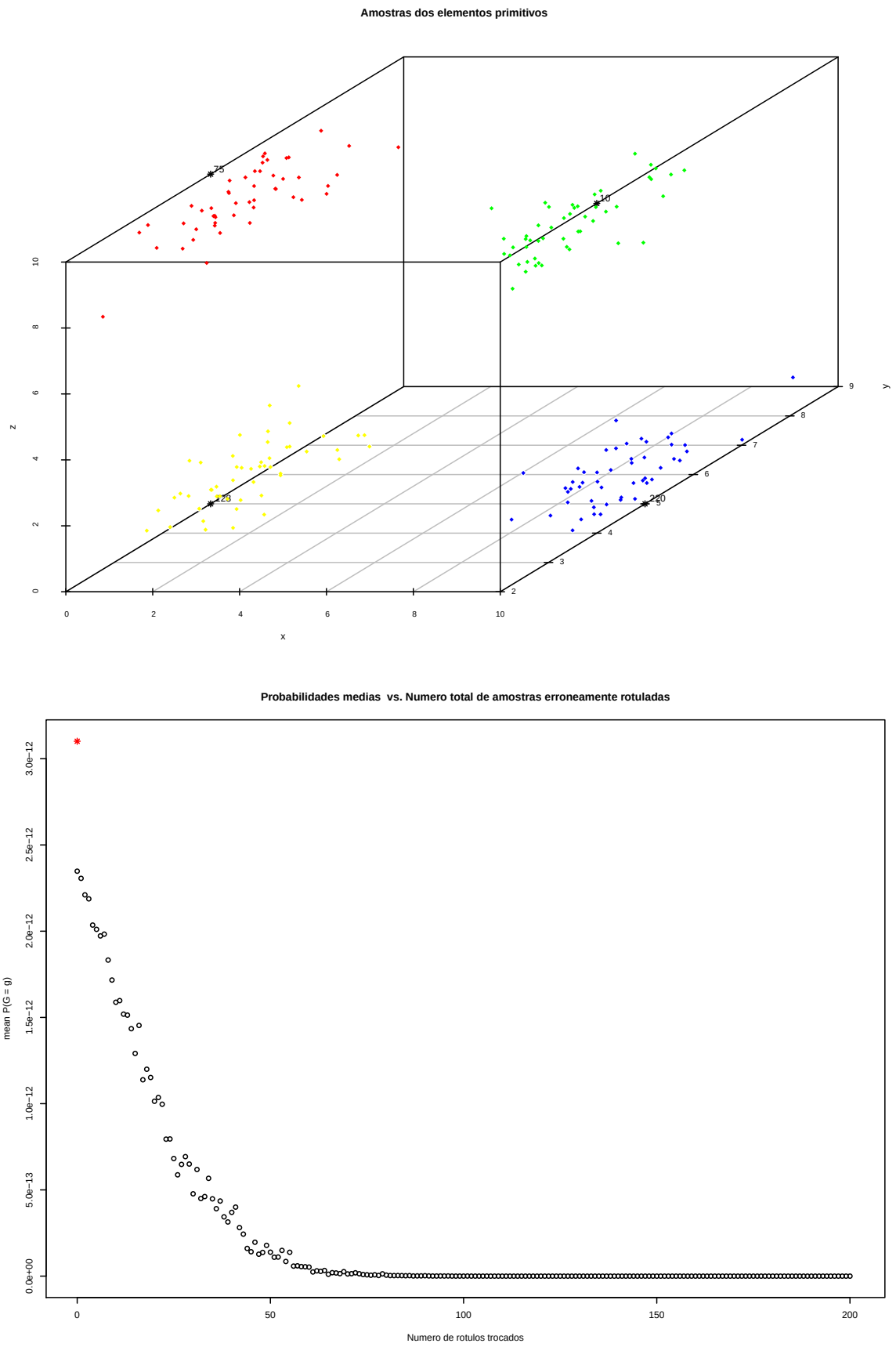


Figura 17: Resultados para Simulação 2: dados amostrados e respectivas curvas probabilidade $\times$ erros para $k = 5$ e $n = 50$.

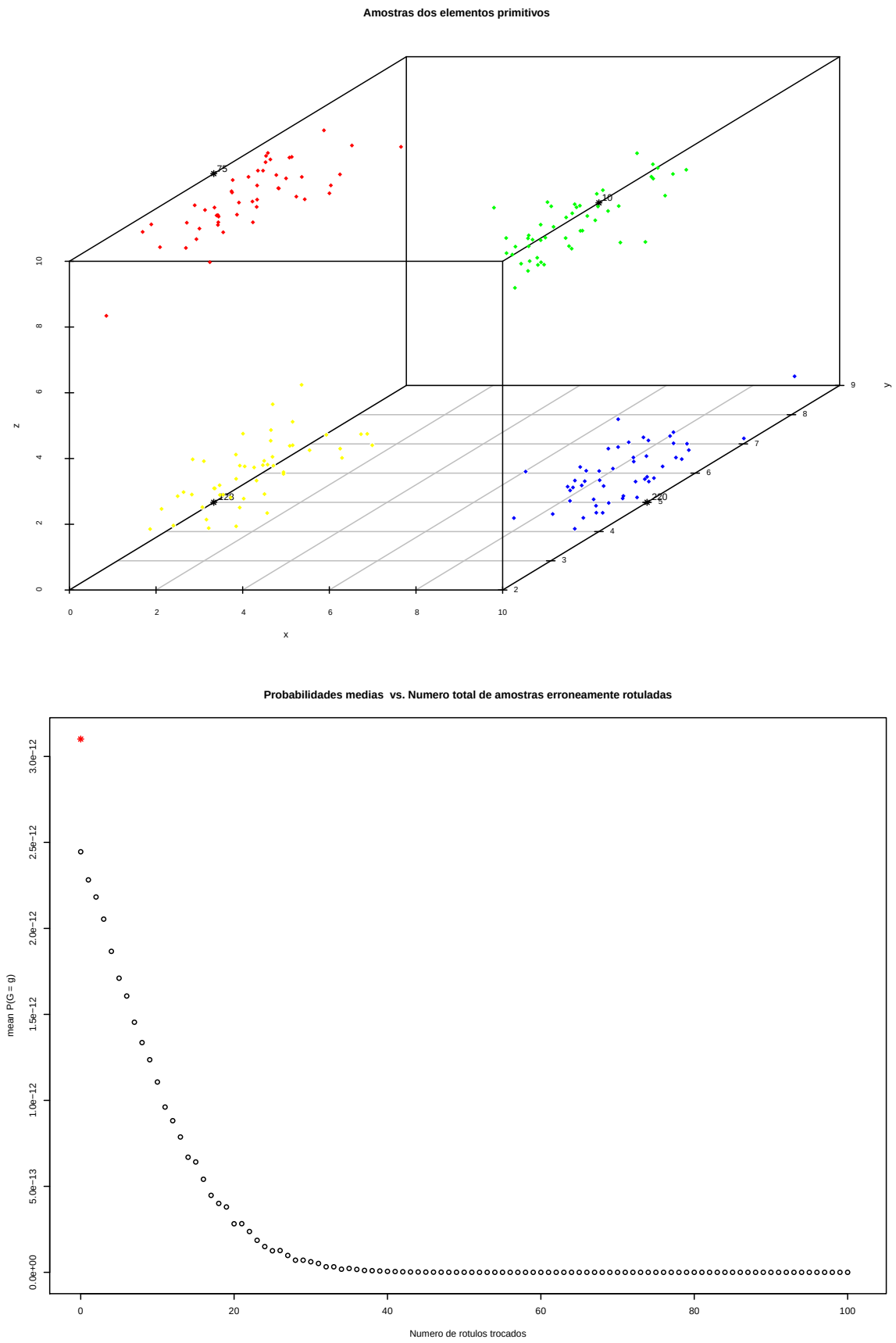


Figura 18: Resultados para Simulação 2: dados amostrados e respectivas curvas probabilidade $\times$ erros para $k = 100$ and $n = 25$.

3.4.5 *Simulação 3*

Objetivo: testar o comportamento de $P(G_m = g_{obs})$ sob a hipótese de independência mútua entre os nós e as variáveis aleatórias que representam atributos e relações. As variáveis aleatórias que representam os níveis de cinza e as coordenadas espaciais dos elementos primitivos apresentam, neste caso, desvios padrões bastante significativos, fazendo com que grande parte de suas respectivas funções de densidade de probabilidade se sobreponham. Dessa forma, a simulação busca representar elementos primitivos com características de aparência e relacionais menos discriminantes, e avaliar a capacidade de detectar esse fenômeno através da probabilidade da observação.

- $\hat{\theta}^{obj}(\overrightarrow{X}(v_m^1)) = [X_1(v_m^1) \sim \mathcal{N}(128, 50), X_2(v_m^1) \sim \mathcal{N}(0, 1),$ $X_3(v_m^1) \sim \mathcal{N}(5, 1), X_4(v_m^1) \sim \mathcal{N}(0, 1)]$;
- $\hat{\theta}^{obj}(\overrightarrow{X}(v_m^2)) = [X_1(v_m^2) \sim \mathcal{N}(220, 30), X_2(v_m^2) \sim \mathcal{N}(5, 1),$ $X_3(v_m^2) \sim \mathcal{N}(5, 1), X_4(v_m^2) \sim \mathcal{N}(6, 1)]$;
- $\hat{\theta}^{obj}(\overrightarrow{X}(v_m^3)) = [X_1(v_m^3) \sim \mathcal{N}(75, 40), X_2(v_m^3) \sim \mathcal{N}(5, 1),$ $X_3(v_m^3) \sim \mathcal{N}(3, 1), X_4(v_m^3) \sim \mathcal{N}(8, 1)]$;
- $\hat{\theta}^{obj}(\overrightarrow{X}(v_m^4)) = [X_1(v_m^4) \sim \mathcal{N}(50, 45), X_2(v_m^4) \sim \mathcal{N}(7, 1),$ $X_3(v_m^4) \sim \mathcal{N}(4, 1), X_4(v_m^4) \sim \mathcal{N}(7, 1)]$;

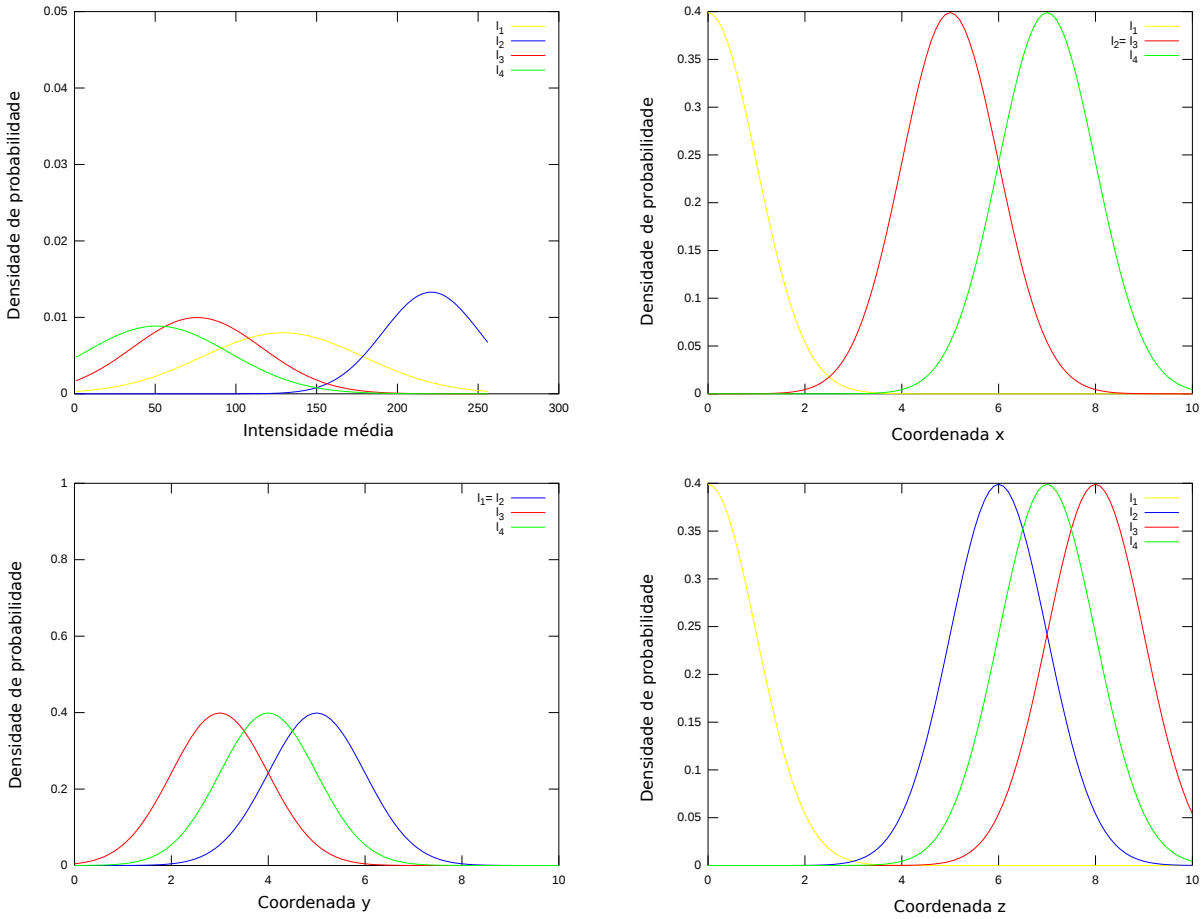
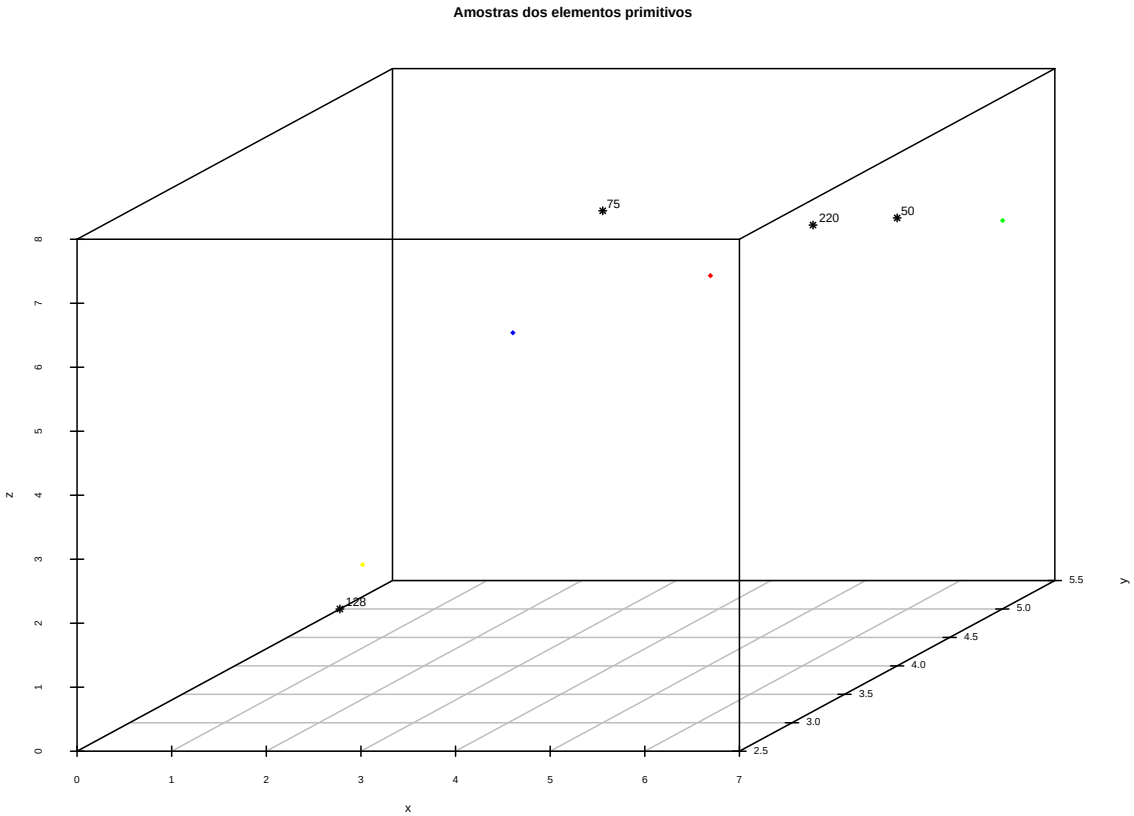


Figura 19: Simulação 3: densidades de probabilidade das variáveis aleatórias representantes das características de aparência de cada elemento primitivo modelado: l_1 (amarelo), l_2 (azul), l_3 (vermelho), l_4 (verde).



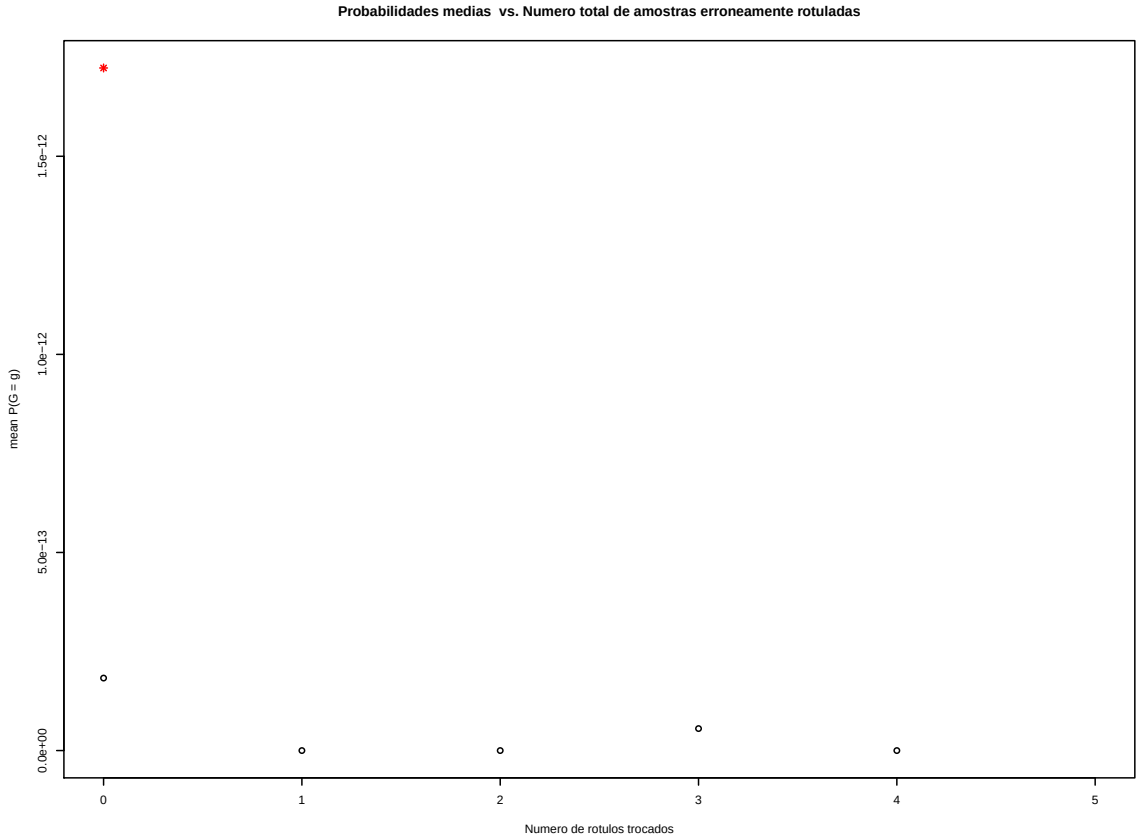


Figura 20: Resultados para Simulação 3: dados amostrados e respectivas curvas probabilidade $\times$ erros para $k = 1$ e $n = 1$.

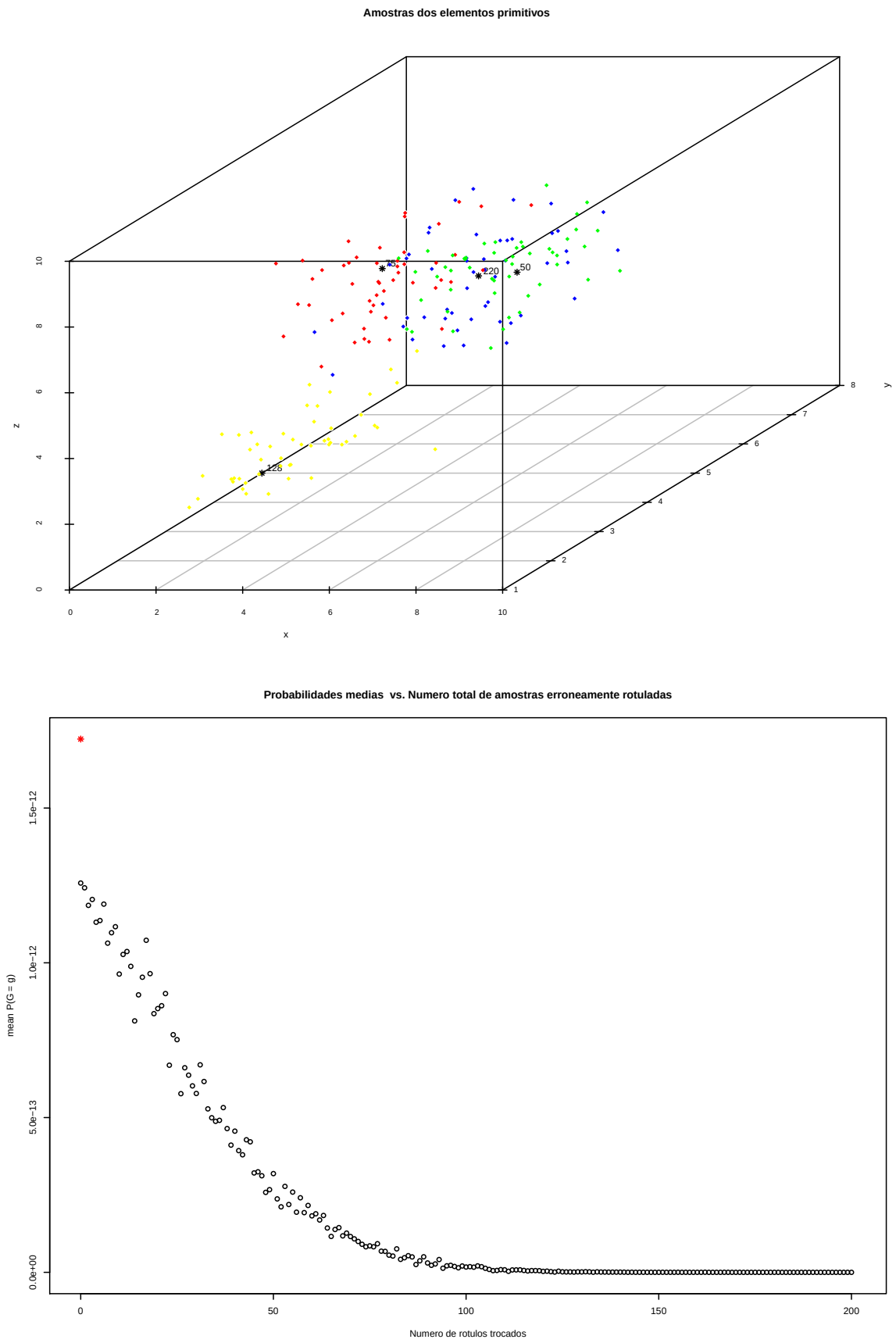


Figura 21: Resultados para Simulação 3: dados amostrados e respectivas curvas probabilidade $\times$ erros para $k = 5$ e $n = 50$.

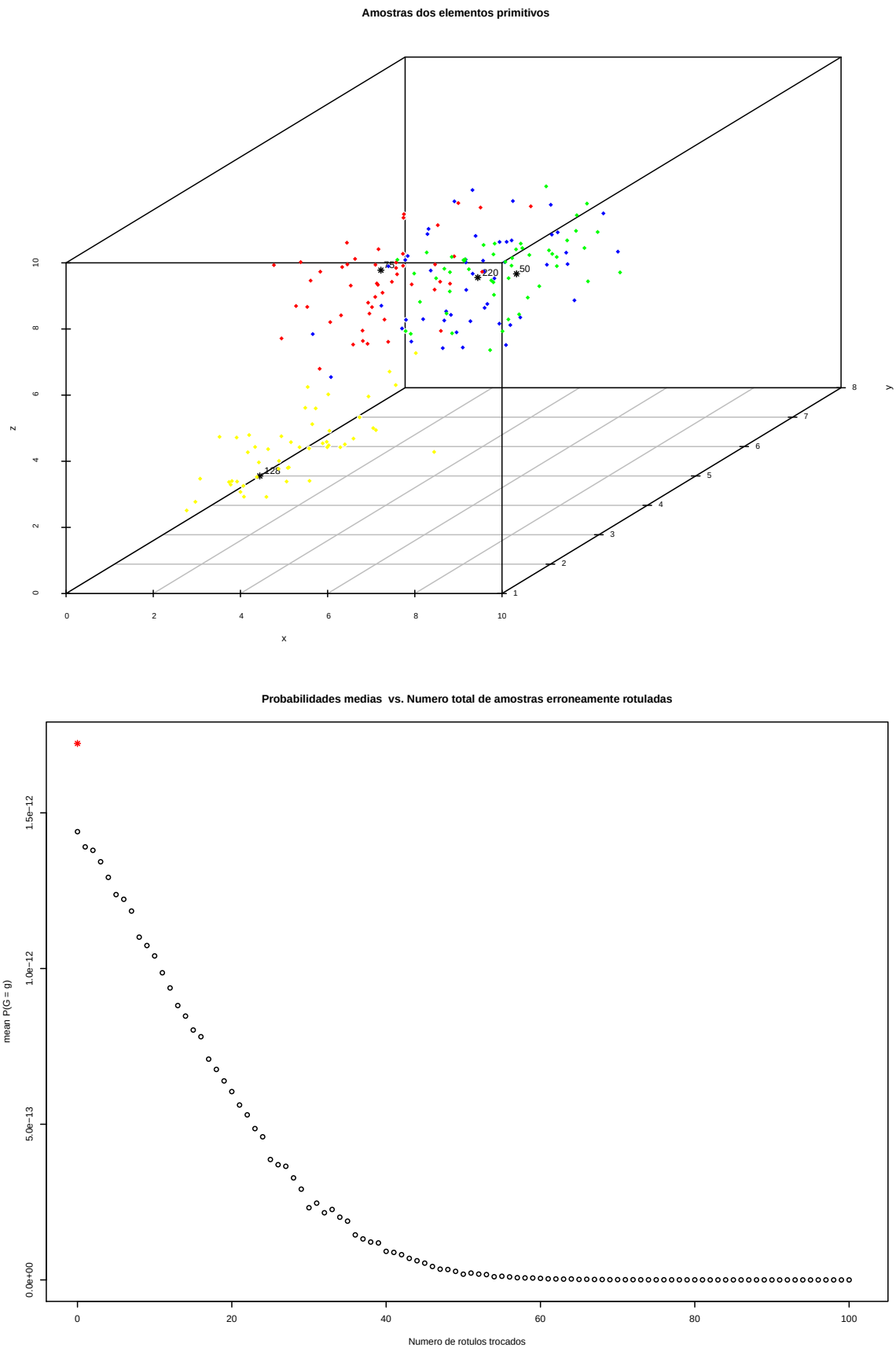


Figura 22: Resultados para Simulação 3: dados amostrados e respectivas curvas probabilidade $\times$ erros para $k = 100$ and $n = 25$.

3.4.6 *Análise das simulações*

Uma análise dos resultados das simulações revela os seguintes aspectos do comportamento probabilístico:

- a melhor rotulação possível (g^*_{obs}), corresponde, de fato, àquela em que cada amostra sorteada recebe o rótulo do elemento primitivo que a gerou e que resulta numa observação cujas medidas equivalem às médias das distribuições das variáveis aleatórias contidas em G_m. A probabilidade mais alta corresponde a esse caso ideal;

- uma outra rotulação correta, em que cada amostra sorteada recebe o rótulo do elemento primitivo que a gerou, mas resulta numa observação com medidas diferentes das médias (o que está mais próximo do que realmente ocorre com imagens verdadeiras), apresenta a maior probabilidade média após aquela do caso ideal.

- conforme o número de erros numa solução aumenta, a probabilidade média para tal número de erros diminui. Logo, soluções incorretas são menos prováveis em relação aos dados transmitidos pelo GER modelo do que soluções que estão mais próximas da rotulação correta. Contudo, esse resultado é válido assintoticamente. Quando o número de erros é pequeno em relação ao número de amostras, as probabilidades não necessariamente decaem monotonicamente.

Em relação à hipótese de independência, observamos que seu comportamento assintótico está de acordo com o esperado, dado que rotulações equivalentes a segmentações muito ruins têm probabilidade menor do que rotulações equivalentes a segmentações melhores. Porém, a probabilidade média não necessariamente decaiu monotonicamente quando o número de erros foi menos expressivo em função do número de amostras. Esse fato pode ser devido à aleatoriedade das amostras que, mesmo rotuladas incorretamente, podem alterar pouco as medidas globais. Ainda, esse fato pode ser devido à escolha dos atributos, que, por serem simples, não discriminam bem as soluções em todas as situações. Contudo, de maneira global, o desempenho do método sob essa hipótese, dada sua simplicidade, pode ser considerado bastante satisfatório, e o comportamento de $P(G_m = g_{obs})$ demonstra seu potencial como função-objetivo.

É importante salientar que a simulação proposta teve por objetivo avaliar a hipótese de independência total mútua sob condições controladas, mas passíveis de ocorrerem em problemas práticos. No Capítulo 4 a seguir, apresentaremos uma aplicação da modelagem por GER ao domínio de imagens de ressonância magnética, para representar

um subconjunto de órgãos abdominais, e discutiremos experimentos sob essa mesma hipótese, realizados com essas imagens.

4 APLICAÇÕES

Eu escuto, e depois esqueço;
eu vejo, e depois me lembro;
eu faço, e então aprendo.

— Confúcio

A metodologia descrita neste documento se destina ao reconhecimento de objetos em imagens de maneira geral, graças à amplitude do conceito de objeto adotado e da representação por grafos estatístico-relacionais, que comporta tanto a modelagem de imagens bidimensionais quanto tridimensionais. No entanto, sua aplicabilidade favorece o tratamento de imagens contendo objetos que podem ser decompostos em partes integrantes, ou conjuntos estruturados compostos por diferentes objetos. Em especial, neste capítulo, discutiremos uma aplicação da metodologia à modelagem e à segmentação de órgãos abdominais em imagens de ressonância magnética (IRM)[1].

4.1 MODELAGEM E SEGMENTAÇÃO DE IMAGENS DE RESSONÂNCIA MAGNÉTICA

O processamento de imagens médicas é uma aplicação de visão computacional que frequentemente inclui o reconhecimento de objetos como um de seus tópicos de pesquisa [MK06, OS01], sobretudo para detectar estruturas anatômicas específicas ou anômalas. Essas imagens possuem características bastante variáveis e dificuldades intrínsecas de processamento de acordo com o método de aquisição (por ex.: tomografia computadorizada, ressonância magnética, ultrassom, etc.), o conteúdo de estudo (partes do corpo humano ou de seres vivos a serem analisadas), as dimensões da imagem adquirida (2D, 3D, vídeo), dentre muitos outros aspectos. As informações extraídas de imagens médicas permitem auxiliar o profissional de medicina a realizar diagnósticos de forma (semi)automática, planejar e efetuar cirurgias, implementar estudos de caráter psico-físico, anatômico, ou descobrir características de patologias diversas, entre outros.

Em geral, os elementos estudados através de imagens médicas, sejam eles macro ou micro elementos do corpo humano, podem apresentar propriedades estruturais e relacionais bastante próprias, ainda

1 A aplicação da metodologia ao domínio de imagens de ressonância magnética e à modelagem do abdome foi desenvolvida durante estágio de doutorado sanduíche junto ao Institut Télécom ParisTech, sob supervisão da Profa. Isabelle Bloch, no período de março/2010 a janeiro/2011.

que com alto grau de variabilidade entre indivíduos diferentes. Portanto, é desejável considerar características relacionais apresentadas entre os órgãos, bem como estatísticas sobre esses elementos, para lidar também com a variabilidade inter-indivíduo.

Diversos estudos concentram-se na segmentação e reconhecimento de estruturas cerebrais [GCBE98], cardíacas e torácicas [NvBF+02]. Porém, comparativamente, poucos trabalhos se dedicam ao processamento de imagens abdominais, cujo estudo permite a criação de métodos não-invasivos para diagnóstico de doenças gastro-intestinais, como a endoscopia virtual [Bar05, WFM+05], diagnóstico de tumores [BRB08], dentre outros avanços importantes para a área de medicina virtual.

O desenvolvimento de técnicas de visão computacional para segmentação de órgãos abdominais em imagens médicas representa um nicho de pesquisa de grande interesse, pois tais técnicas são uma etapa importante constituinte de várias aplicações médicas [MK06]. Particularmente, grande parte dos métodos destinados à segmentação do abdome está voltada para o tratamento de imagens provenientes de tomografia computadorizada (TC) [Var02, CCP10]. Contudo, é interessante explorar também a segmentação de imagens obtidas por ressonância magnética (RM). Além do caráter não-radioativo, essa modalidade de imageamento médico apresenta vantagens sobre imageamento por TC em certas circunstâncias como, por exemplo, no diagnóstico de tumores, alterações no fígado, na vesícula, ou nos rins, além de detecção de abscessos na região abdominal.

Contudo, a segmentação manual de órgãos requer bons conhecimentos anatômicos por parte do usuário que realiza a segmentação (por ex., o radiologista). Além disso, ela é uma tarefa exaustiva, dado que os órgãos a serem segmentados aparecem em numerosos cortes da imagem. Portanto, o desenvolvimento de métodos (semi-)automáticos para segmentação de imagens RM pode auxiliar no diagnóstico ou tratamento de diferentes patologias. Além disso, técnicas de segmentação auxiliadas por computador facilitariam a utilização de grande volume de dados de imageamento médico na criação de modelos, ou atlas, computacionais.

Do ponto de vista técnico, a segmentação automática de órgãos abdominais em imagens é um desafio devido tanto à variabilidade anatômica e funcional que lhes é inerente, quanto às especificidades dos equipamentos de ressonância magnética. Órgãos podem apresentar variabilidade inter-indivíduo em termos de tamanho e formato, ou intra-indivíduo, quando são comparadas aquisições distintas de um mesmo paciente em momentos diversos de sua vida. Um mesmo órgão também pode ser caracterizado por diferentes radiometrias em imagens distintas, segundo o seu conteúdo específico quando da aquisição (por ex.: água, ar, gordura, etc.). Outras variações radiométricas entre imagens podem ser causadas por aquisições feitas com equi-

pamentos regulados de forma distinta, ou que pertencem a modelos diferentes (fabricante ou série), ou, ainda, devido à degradação natural sofrida por um equipamento ao longo do tempo.

Toda essa variabilidade anatômica e de aquisição de imagens afeta diretamente o sucesso dos métodos automáticos para segmentação, sobretudo aqueles que se apóiam em características de aparência dos órgãos [CGX08, Jö06, LHS08, WPNK08]. Embora essas propriedades sejam relevantes [Kar90], outras fontes de conhecimento a priori podem ser exploradas na descrição dos órgãos. Particularmente, os órgãos abdominais apresentam um arranjo espacial padrão que permite situá-los individualmente em relação às posições de outros órgãos. Além disso, os rins, por exemplo, apresentam um certo grau de simetria bilateral. Essas relações comuns aos órgãos da maioria dos seres humanos podem auxiliar o processo de segmentação de imagens e torná-lo mais robusto. Na literatura específica, ainda são poucos os trabalhos que investem nas características relacionais entre os órgãos [PBM03, ZB07].

Todas essas fontes de informação sobre os órgãos são relevantes para o problema de segmentação automática de órgãos abdominais, e a combinação de vários desses aspectos num único formalismo seria de grande valia. Esse fato vem ao encontro da metodologia baseada em modelo descrita no capítulo precedente. Portanto, neste capítulo descreveremos a aplicação desse formalismo ao processamento de imagens de ressonância magnética não-contrastadas da região abdominal, visando prover um modelo simplificado do sistema gastrointestinal representado por um grafo estatístico-relacional, bem como estudar sua aplicabilidade ao reconhecimento dos órgãos de interesse em novas imagens.

Essa proposta define dois problemas a serem tratados: criação de um grafo estatístico-relacional (GER) modelo representante dos órgãos, seguida de segmentação desses órgãos numa nova imagem segundo o modelo. A etapa de obtenção do grafo modelo consiste no aprendizado supervisionado dos atributos desse grafo a partir de um conjunto de imagens de treinamento em que os órgãos de interesse foram previamente segmentados e rotulados à mão e validadas por um especialista médico.

Numa segunda etapa, dada uma imagem de RM da cavidade abdominal, obtém-se uma segmentação inicial por regiões (subconjuntos de voxels) dessa imagem e um GRA é extraído desse particionamento. A segmentação dos órgãos de interesse na imagem de entrada é derivada da classificação das regiões representadas pelo GRA de acordo com as características e relações contidas no GER modelo. O casamento entre os grafos, que corresponde a uma segmentação, é então interpretado de uma maneira probabilística, em que verificamos a hipótese do GRA observado ser uma realização do GER modelo.

O caráter híbrido (estatístico-estrutural) da modelagem proposta permite o uso efetivo de relações entre propriedades dos órgãos, ou de relações espaciais, na etapa de classificação, o que pode ser decisivo no caso de imagens médicas, dado que fatores estruturais e relacionais também são bastante discriminantes. Ao mesmo tempo, são consideradas as informações estatísticas de propriedades dos órgãos, como se faz nas abordagens puramente estatísticas. Além disso, a representação das imagens em função de regiões (subconjuntos de voxels) é mais compacta do que aquela baseada em unidades (voxel-a-voxel), permitindo também a formulação de hipóteses mais elaboradas e o raciocínio sobre informações de mais alto nível em termos de relações espaciais e características de forma ou de objeto. Já no processamento baseado em elementos pictóricos individuais, a conectividade se restringe às relações de vizinhança da grade que define a imagem, enquanto as características de aparência dos elementos restringem-se praticamente à intensidade.

Todos esses aspectos da metodologia serão descritos no decorrer deste capítulo.

4.2 DADOS DE RESSONÂNCIA MAGNÉTICA ABDOMINAL

Nesta aplicação voltada para o tratamento de imagens médicas, utilizamos imagens de ressonância magnética não-contrastadas, obtidas durante exames clínicos de rotina de 10 pacientes com idades entre 10 e 17 anos. Todos esses dados foram obtidos por protocolos de aquisição do tipo T1, e cada volume apresenta resolução de 1.3mm $\times$ 7.2mm $\times$ 1.3mm, nas direções axial, coronal e sagital, respectivamente. A Figura 23 apresenta alguns cortes dessas imagens compreendendo o abdome.

Todas as imagens foram concedidas pela equipe de radiologia pediátrica do Hospital St. Vincent de Paul, em Paris, no âmbito de uma cooperação com o projeto *KidPocket* desenvolvido pela France Télécom em parceria com a Télécom ParisTech. Esse projeto prevê a modelagem 3D do corpo humano, por computação gráfica, para avaliar os possíveis efeitos biológicos induzidos por ondas eletromagnéticas provenientes de dispositivos móveis, como aparelhos celulares. O projeto já cobriu estudos com enfoque sobre o cérebro humano, pacientes grávidas e seus fetos, e prevê o abdome em sua continuação. A construção dessa representação 3D por computação gráfica se apóia em imagens médicas de pacientes reais, previamente segmentadas em elementos de interesse para o modelo 3D, como ossos, pele, órgãos, entre outros. Portanto, a segmentação de órgãos abdominais proposta mostra-se útil também no contexto desse projeto.

A partir dessas imagens, podemos criar um modelo computacional do abdome, para utilizá-lo dentro do paradigma de segmentação de imagens baseada em modelo.

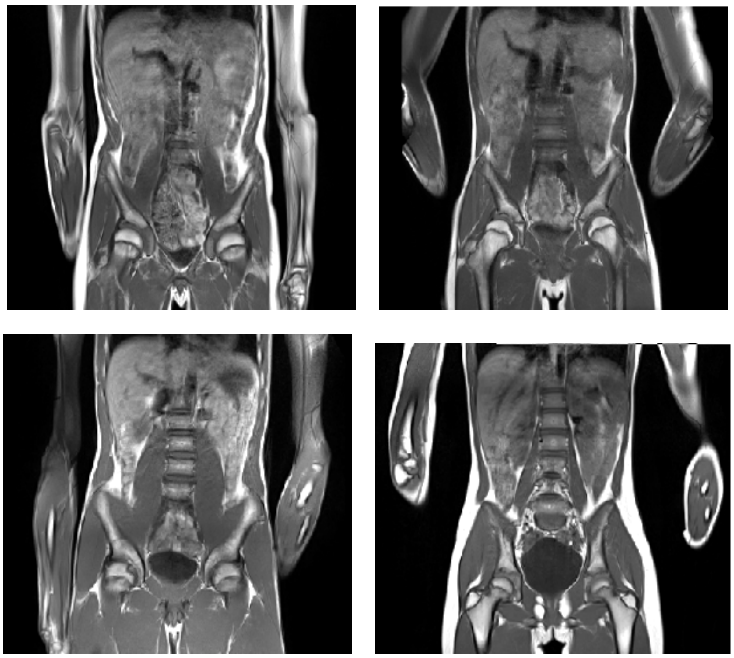


Figura 23: Imagens de RM de pacientes distintos, obtidas durante protocolos de aquisição para o corpo inteiro, porém com volumes restritos à cavidade abdominal e à região pélvica de interesse. Note que os mesmos órgãos aparecem com características variadas, de acordo com as especificidades anatômicas de cada paciente, do conteúdo do órgão no instante da aquisição (por ex.: ar, líquido) ou mudanças de forma devido à elasticidade do órgão (por ex.: estômago/bexiga vazio/a ou cheio/a), dentre outros fatores.

4.3 O MODELO ABDOMINAL

4.3.1 *Definições e notações*

Ao longo deste capítulo, reutilizaremos os conceitos e notações descritos na Seção 3.1. Contudo, recapitulamos alguns deles explicitamente, para precisar o sentido que assumem na aplicação de IRM.

Considere P a representação digital de uma imagem tridimensional de ressonância magnética, e R o seu domínio, segundo a descrição em $\mathbb{N}^3_+$. O número total de voxels da imagem será dado por $|R| = w \times h \times z$, sendo w o número de colunas da imagem, h o seu número de linhas, e d sua profundidade.

$I : R \mapsto \mathbb{R}^+$ denota a função de radiometria associada a um voxel de P, i.e., $I(x, y, z)$ é o valor da radiometria quantizada para o voxel dado pelas coordenadas x, y, z. Finalmente, L representa o conjunto de rótulos representantes de órgãos abdominais a serem modelados, e ρ, um rótulo associado a todos os órgãos não representados em L. Ainda, $L^* = L \cup \{\rho\}$.

4.3.2 *Descrição conceitual*

Um primeiro passo para a criação de um modelo computacional do abdome é definir um modelo conceitual para descrever os dados de ressonância magnética. Em primeiro lugar, é necessário selecionar os órgãos que deverão ser objeto do estudo em questão. Adotamos $L = \{$fígado, estômago, coluna vertebral, rim direito, rim esquerdo$\}$ como o conjunto de órgãos a serem modelados. Decidimos criar esse modelo simplificado representando apenas esses órgãos, pois os mesmos estavam presentes e visíveis em todas as imagens disponíveis. Demais órgãos abdominais, tais como pâncreas e duodeno, não foram modelados, pois não apareciam em todas as imagens devido à resolução das mesmas. Apesar da coluna não ser uma víscera abdominal, ela é sempre vísivel em ressonâncias magnéticas abdominais e constitui uma estrutura relevante a partir da qual derivam várias relações espaciais, daí a sua presença no modelo.

Embora o objetivo seja modelar os órgãos de interesse do abdome, é necessário considerar também as relações que esses têm com outras regiões do corpo circunvizinhas e que não são modeladas detalhadamente. Para incorporar essas regiões ao modelo, introduzimos um elemento genérico denominado *fundo*, representado pelo símbolo ρ. Logo, $L^* = L \cup \{\rho\}$ consiste no conjunto final de órgãos (classes) de interesse.

O conhecimento espacial acerca do arranjo dos órgãos de interesse é representado pela rede semântica exibida na Figura 24. Como o fundo circunda todos os órgãos, ainda que de maneiras distintas, ele foi omitido da rede apresentada nessa mesma figura.

4.3.3 *Propriedades dos órgãos abdominais e suas relações*

Uma vez selecionado o conjunto de órgãos a modelar, diversas propriedades podem ser derivadas. A descrição proposta para cada órgão é baseada na inspeção visual e na análise qualitativa das imagens disponíveis, bem como nas informações anatômicas compiladas em [Val03], ou ontologias existentes para o corpo humano, tais como o Modelo Fundamental de Ontologia Anatômica (Foundational Model of Anatomy Ontology - FMA) [FMA95] e Galen [Ope10].

FÍGADO: o fígado é um órgão de destaque no abdome, pois é a maior glândula do corpo humano. É composto de um grande lóbulo direito e outro menor esquerdo. Seu tecido é predominantemente composto de água, cuja porcentagem varia conforme a faixa etária do indivíduo (em média 71% na idade adulta e 73% na infância). Normalmente, o fígado está localizado no quadrante superior direito da cavidade abdominal e sua extensão abrange tanto as partes anterior quanto posterior.

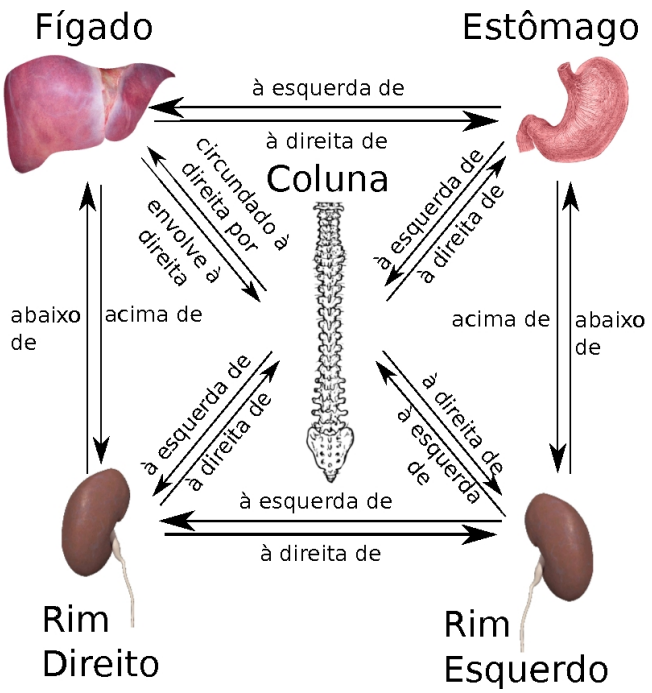


Figura 24: Uma rede semântica para modelar os órgãos abdominais de interesse.

ESTÔMAGO: esse órgão funciona como uma bolsa vazia que se distende de acordo com a quantia de alimento ingerido, o que ocasiona alta variabilidade em termos de forma e dimensões. Além desse fator, os movimentos peristálticos, a contração muscular própria do músculo, e também a das paredes abdominais, podem influenciar sua aparência. Contudo, habitualmente ele está localizado anteriormente no quadrante superior esquerdo da cavidade abdominal e apresenta a forma de um gancho quando visto em relação a uma pessoa em pé.

RINS DIREITO E ESQUERDO: esses dois órgãos têm forma similar a de um grão de feijão, e suas dimensões são próximas às de uma mão bem fechada. Seus tecidos são compostos majoritariamente de água (76% em média). Ambos estão localizados na parte posterior do abdome, estando o rim direito abaixo do fígado e ligeiramente deslocado para baixo em relação ao rim esquerdo.

COLUNA VERTEBRAL: graças a sua posição axial no corpo, a coluna vertebral pode ser facilmente relacionada espacialmente com outros órgãos para determinar suas posições.

De acordo com essas descrições, podemos derivar medidas para representar essas características próprias da aparência de cada órgão, ou das relações entre eles.

Em particular, para os atributos de aparência, adotamos as seguintes medidas:

- radiometria média;
- volume do órgão;
- coordenadas do centróide.

A radiometria da imagem RM pode ser associada a um órgão graças a sua permeabilidade a um dado tipo de meio orgânico (água, ar), pois, dependendo do protocolo utilizado, sabe-se que esses meios tendem à aparência mais clara, ou mais escura. Logo, órgãos compostos majoritariamente de moléculas de água estarão associados a uma radiometria distinta de outros tais como a coluna, composta de material ósseo.

Contudo, não existe uma escala de radiometria padrão para cada órgão quando se comparam imagens de ressonância magnética obtidas através de aquisições ou protocolos distintos em geral. Isso dificulta a associação de cada órgão a uma faixa radiométrica específica para imagens inter-indivíduo, ou mesmo para imagens intra-indivíduo.

Para amenizar esse problema, faz-se necessário normalizar a radiometria das imagens, por exemplo, através do aprendizado de um histograma padrão para um dado protocolo de aquisição e subsequente mapeamento das imagens para esse histograma, conforme descrito em [JNF+07].

Por sua vez, o volume de um órgão permite distinguí-lo dos demais de acordo com a escala de tamanho em que se encontra, ainda que essa medida não seja tão expressiva para órgãos muito deformáveis, como o estômago. Já o centróide de um órgão permite exprimir a sua localização aproximada na imagem.

Quanto ao conjunto de relações verificáveis entre os órgãos pertencentes a L, selecionamos os seguintes atributos relacionais:

- contraste entre as intensidades médias de um par de órgãos;
- proporção entre volumes;
- distância euclidiana entre centróides.

O contraste é calculado com base na radiometria média de um órgão, e se comporta de maneira satisfatória para órgãos relativamente homogêneos. Uma outra possibilidade para tratar órgãos como o estômago, que pode ter aparência variável conforme seu conteúdo, seria o cálculo de contraste entre as superfícies externas de dois órgãos. A proporção permite avaliar o par de órgãos em questão de maneira independente do tamanho do volume que os contém. Esse fator permite comparar proporções calculadas sobre imagens de pacientes distintos, independentemente de suas alturas. Finalmente, a distância

euclidiana entre os centróides é uma medida compacta e simples de se calcular, que fornece dados acerca do arranjo espacial dos órgãos. Embora possa haver dois órgãos distintos com mesma distância em relação a um terceiro órgão, essa medida, em conjunto com as demais, permite caracterizar corretamente diferentes pares de órgãos. Uma medida de distância entre superfícies também poderia ser adotada, se a caracterização por distância euclidiana não fosse suficiente.

É importante observar que qualquer uma dessas características, tomadas de modo isolado, podem ser similares para órgãos distintos (por exemplo: radiometria dos rins direito e esquerdo), ou relações distintas (por exemplo: proporção entre cada rim e a coluna vertebral). Daí a importância da utilização conjunta dos atributos de aparência e relacionais, que serve como mecanismo regulador na segmentação de um dado órgão, observando suas interações com os demais.

Nas seções seguintes, discutiremos como representar essas informações anatômicas computacionalmente e utilizá-las na segmentação supervisionada de imagens RM conforme o arcabouço descrito no Capítulo 3.

4.4 SEGMENTAÇÃO DE ÓRGÃOS ABDOMINAIS EM IRM

A segmentação de uma imagem P de ressonância magnética do abdome segundo um modelo pode ser interpretada dentro do contexto da abordagem estatístico-estrutural apresentada no Capítulo 3.

De acordo com essa formulação, o modelo conceitual é traduzido num *grafo estatístico-relacional*, que descreve tanto as informações estatísticas, quanto o conhecimento anatômico sobre o arranjo espacial do sistema estruturado de órgãos discutido na Seção 4.3.3. Neste caso, o GER modelo é um grafo relacional com atributos em que nós representam os órgãos de interesse em L^*, e arcos indicam as relações entre esses órgãos. Além disso, cada nó ou arco está associado a vetores de atributos que representam os parâmetros das distribuições de probabilidade estimados para as características e relações descritas na Subseção 4.3.3.

Por outro lado, consideramos uma segmentação da imagem P, dada por S e f, como uma possível observação do modelo abdominal, a qual é representada por um grafo relacional com atributos, denominado *grafo de observação*. Nesse grafo, os atributos são observações das variáveis aleatórias referentes às propriedades e relações do modelo.

Com base nessas representações, podemos então avaliar essa segmentação dada para P segundo o modelo estatístico-relacional, de modo probabilístico.

4.4.1 *Representação de uma segmentação por grafo de observação*

Considere uma partição S de P em regiões disjuntas, obtida através de algoritmos de segmentação 3D tais como os algoritmos de linha de partição de águas (*watershed*), ou *quad-trees*. Nesses casos, cada s_i corresponde a uma região disjunta produzida por um tal algoritmo. Por ex.: cada s_i corresponde a uma bacia de captação de águas proveniente de uma segmentação por linha de partição de águas do gradiente de uma imagem P. A essa partição, associamos um mapeamento f, que corresponderá à rotulação de cada s_i de acordo com os elementos em L^*, segundo algum critério de inicialização.

Com base em S e f, calculamos as medidas observadas e equivalentes às propriedades de aparência e relacionais de cada órgão de interesse (Seção 4.3.3) em P. Assim, cada órgão está associado a um vetor aleatório que corresponde às seguintes observações:

- radiometria média, denotada por x_1;
- volume do órgão, denotado por x_2.
- coordenadas do centróide, denotadas por x_3^x, x_3^y, e x_3^z.

Além disso, cada relação presente entre órgãos está associada a um vetor aleatório que comporta as seguintes observações:

- contraste entre as intensidades médias de um par de órgãos, denotada por y_1;
- proporção entre áreas, denotada por y_2;
- distância entre centróides, denotada por y_3.

A partir desses elementos, definimos um *grafo de observação* $g_{obs} = (V_{obs}, E_{obs}, \overrightarrow{x}, \overrightarrow{y})$, em que:

- V_{obs} é o conjunto de nós de g_{obs}. Cada nó v^i representa um elemento $l \in L$ (um órgão, ou o fundo ρ) e as respectivas regiões expandidas s_l de S. Assumimos $|V_{obs}| = |L^*|$ para aplicar o procedimento de reconhecimento estatístico-estrutural posteriormente;
- E_{obs} é o conjunto de arcos de g_{obs}. Um arco é criado entre os vértices representantes de dois conjuntos de regiões expandidas, representando órgãos que estão relacionados em G_m;
- $\overrightarrow{x}(v^i) = [x_1(v^i), \ldots, x_{v_{V_m}}(v^i)]$ é uma função que devolve o vetor de valores observados x_p, $p = 1, \ldots, v_{V_m}$, relativos às variáveis aleatórias X_i que descrevem as propriedades dos órgãos do modelo. Esses valores são calculados para o conjunto de regiões expandidas correspondentes ao nó v^i;

- $\overrightarrow{y}(e^{ij}) = [y_1(e^{ij}), \ldots, y_{v_{E_m}}(e^{ij})]$ é uma função que devolve o vetor de valores observados y_q, $q = 1, \ldots, v_{E_m}$, relativos às variáveis aleatórias Y_j que descrevem as relações entre os órgãos do modelo. Esses valores são calculados para o par de conjuntos de regiões expandidas representadas pelos nós conectados por e^{ij}.

Calculando atributos de aparência e relacionais

A partir de I, de S, e de f, podemos medir os valores observados para cada variável aleatória representando atributos de aparência de um dado órgão $l \in L^*$ na imagem P, representado por um nó $v \in |V_{obs}|$. Ainda, podemos medir também os valores observados para cada variável aleatória representando atributos relacionais de um par de órgãos em L^*, representados por v^i e v^j, cujas relações são expressas por e^{ij}. Todas essas medidas serão armazenadas nos vetores de atributos associados ao nó v e ao arco e^{ij} de g_{obs}.

Para calcular as observações referentes aos atributos de aparência de v, procedemos conforme a descrição a seguir.

RADIOMETRIA MÉDIA. Para medir a radiometria média x_1 de um conjunto de regiões contidas em s_l, i.e., regiões cujos voxels foram rotulados como pertinentes ao órgão l numa dada segmentação S, simplesmente calculamos a média aritmética de suas radiometrias.

$$x_1(v) = \frac{1}{\sum_{s_i \subset s_l} |s_i|} \sum_{(x,y,z) \in s_i, s_i \subset s_l} I(x,y,z) \tag{4.1}$$

TAMANHO. O tamanho de um órgão é proporcional ao número de voxels das regiões contidas em s_l para uma dada segmentação, de acordo com as dimensões do voxel. Portanto, de modo simplificado, tomamos x_2 como sendo a soma das cardinalidades dos conjuntos contidos em s_l:

$$x_2(v) = \sum_{s_i \subset s_l} |s_i|. \tag{4.2}$$

CENTRÓIDE. As coordenadas do centróide de um órgão l são medidas a partir das coordenadas de todos os voxels pertencentes às regiões contidas no respectivo conjunto s_l. O cálculo da coordenada "x" do centróide (x_3^x) é dado por:

$$x_3^x(v) = \frac{1}{\sum_{s_i \subset s_l} |s_i|} \sum_{(x,y,z) \in s_i, s_i \subset s_l} x \tag{4.3}$$

As medidas referentes às demais coordenadas são calculadas de modo análogo, substituindo-se os valores de x por y e z respectivamente.

Já os valores observados para as variáveis aleatórias que representam os atributos relacionais para e^{ij} são calculados como descrito a seguir.

CONTRASTE. Medimos o valor do contraste de Michelson [Pel90] (y_1) entre v^i e v^j através da seguinte fórmula:

$$y_1(e^{ij}) = \frac{|x_1(v^i)-x_1(v^j)|}{x_1(v^i)+x_1(v^j)} \tag{4.4}$$

PROPORÇÃO. Medimos a proporção y_2 entre v^i e v^j através da seguinte fórmula:

$$y_2(e^{ij}) = \begin{cases} x_2(v^i)/x_2(v^j) & \text{se } x_2(v^i) \leqslant x_2(v^j), \\ x_2(v^j)/x_2(v^i) & \text{c.c.}. \end{cases} \tag{4.5}$$

DISTÂNCIA EUCLIDIANA. Medimos a distância Euclidiana (y_3) entre os centróides de v^i e v^j através da seguinte fórmula:

$$y_3(e^{ij}) = \sqrt{dx^2 + dy^2 + dz^2}, \tag{4.6}$$

em que $dx = |x_3^x(v^i) - x_3^x(v^j)|$, $dy = |x_3^y(v^i) - x_3^y(v^j)|$, and $dz = |x_3^z(v^i) - x_3^z(v^j)|$. Os valores $x_3^x(v^i)$, $x_3^y(v^i)$, e $x_3^z(v^i)$ são as respectivas coordenadas do centróide obtido para v^i. O mesmo se aplica às coordenadas do centróide de v^j.

4.4.2 *Representação computacional do modelo conceitual*

De acordo com a modelagem estatístico-relacional, um GER modelo é uma representação para os elementos que compõem um sistema estruturado de objetos, que, neste caso, são os órgãos abdominais definidos em L^*, bem como dados estatísticos acerca das propriedades de aparência e de relações definidas para esses órgãos.

Na clássica abordagem de reconhecimento estatístico de padrões [JDM00], cada órgão de interesse corresponde a uma classe, enquanto suas características são traduzidas em variáveis aleatórias que podem ser observadas a partir de cada exemplar da classe. A variabilidade de expressões de uma classe é, então, expressa por funções de probabilidade (distribuições/densidades), as quais podem ser estimadas a partir de dados de treinamento representativos.

No caso do GER modelo, cada vértice representante de um órgão equivale a uma dessas classes. Porém, o caráter relacional apresentado pelos órgãos é expresso pela topologia do GER, que segue aquela

da rede semântica que descreve as inter-relações dos órgãos, como já mostramos na Figura 24.

As propriedades de um órgão são consideradas como variáves aleatórias, para as quais adotamos a hipótese de que seguem uma distribuição de probabilidade Normal [2]. Portanto, de acordo com as constatações feitas na seção 4.2, as características das regiões da imagem que compõem um órgão (Seção 4.3.3) são dadas pelas seguintes variáveis aleatórias:

- radiometria média, denotada por $X_1 \sim \mathcal{N}(\mu_1, \sigma_1)$;
- volume do órgão, denotado por $X_2 \sim \mathcal{N}(\mu_2, \sigma_2)$.
- coordenadas do centróide, denotadas por $X_3^x \sim \mathcal{N}(\mu_3^x, \sigma_3^x)$, $X_3^y \sim \mathcal{N}(\mu_3^y, \sigma_3^y)$, e $X_3^z \sim \mathcal{N}(\mu_3^z, \sigma_3^z)$.

Para tratar também o aspecto relacional, associamos variáveis aleatórias a cada relação entre órgãos com base nas imagens RM. Supomos que cada variável aleatória segue uma distribuição de probabilidade Normal e representa uma das propriedades relacionais relatadas na Seção 4.3.3:

- contraste entre as intensidades médias de um par de órgãos, denotada por $Y_1 \sim \mathcal{N}(\mu_1^{rel}, \sigma_1^{rel})$;
- proporção entre áreas, denotada por $Y_2 \sim \mathcal{N}(\mu_2^{rel}, \sigma_2^{rel})$;
- distância entre centróides, denotada por $Y_3 \sim \mathcal{N}(\mu_3^{rel}, \sigma_3^{rel})$.

A partir desses elementos, podemos definir formalmente o grafo estatístico-relacional como uma tupla $G_m = (V_m, E_m, \hat{\theta}^{obj}, \hat{\theta}^{rel})$ tal que:

- V_m é o conjunto de nós do grafo G_m. Cada nó representa um único órgão pertencente a L^* e $|V_m| = |L^*|$;
- E_m é o conjunto de arcos do grafo G_m. Cada arco representa as relações existentes entre órgãos pertencentes a L^*;
- $\hat{\theta}^{obj}(\overrightarrow{X}(v_m)) = [\hat{\theta}(X_1(v_m)), \ldots, \hat{\theta}(X_v(v_m))]$, $v_m \in V_m$ é um mapeamento do vetor de variáveis aleatórias $\overrightarrow{X}$ associadas a um nó v_m para um vetor de parâmetros estimados das distribuições referentes às variáveis aleatórias que representam as propriedades do órgão modelado pelo nó v_m. O símbolo v_{V_m} denota o número de propriedades de órgãos consideradas;

2 Para órgãos cujo aspecto varia conforme o seu conteúdo, uma distribuição bimodal pode ser mais adequada.

- $\hat{\theta}^{rel}(\overrightarrow{Y}(e_m)) = [\hat{\theta}(Y_1(e_m)), \ldots, \hat{\theta}(Y_{\nu_{E_m}}(e_m))]$, $e_m \in E_m$ é um mapeamento do vetor de variáveis aleatórias $\overrightarrow{Y}(e_m)$ de um arco e_m para um vetor de parâmetros estimados das distribuições referentes às variáveis aleatórias representantes das relações partilhadas entre os órgãos modelados pelos vértices conectados pelo arco e_m. O símbolo ν_{E_m} denota o número de relações consideradas.

Aprendizado supervisionado do modelo

Os parâmetros das distribuições de probabilidade que regem as variáveis aleatórias associadas aos atributos dos nós e arcos de G_m são estimados através do aprendizado supervisionado baseado num conjunto de imagens de treinamento de ressonância magnética do abdome, nas quais os órgãos de interesse foram manualmente segmentados e rotulados por um especialista capacitado a reconhecer os órgãos de interesse pertencentes a L^*.

Seja esse conjunto de treinamento formado por N imagens $P_1, P_2, \ldots P_N$. Cada imagem de treinamento deve ser segmentada manualmente, de modo que suas respectivas segmentações (S_1, f_1), (S_2, f_2),..., (S_N, f_N) representem de fato uma partição e rotulação corretas dos órgãos de interesse, ou seja, que sejam ideais do ponto de vista de um especialista médico.

Para cada segmentação correspondente a uma imagem de treinamento, calculamos um respectivo GRA $g^1_{obs}, \ldots, g^N_{obs}$. Esses grafos são obtidos da mesma forma que os grafos de observação, e o cálculo de seus vetores aleatórios é realizado da maneira relatada na Seção 4.4.1. Assim, temos um conjunto de grafos de observação que serão utilizados no treinamento e aprendizado do modelo.

A partir desses dados, podemos estimar a média e o desvio padrão das distribuições normais que modelam as variáveis aleatórias associadas aos atributos de G_m. Esse aprendizado é feito através da estimação da máxima-verossimilhança [DHS00] desses parâmetros, calculada a partir das medidas extraídas desse conjunto de amostras. A seguir, vamos mostrar como estimar os parâmetros para cada atributo específico de órgão ou relacional.

Considere um órgão l representado por um nó $v_m \in G_m$, assim como cada nó correspondente $v_k \in V^k_{obs}$, $k = 1, \ldots, N$. Os parâmetros das distribuições relativas aos seus atributos de aparência são calculados conforme descrito a seguir.

RADIOMETRIA MÉDIA. Estimamos os parâmetros da população Normal associada à variável aleatória (X_1) representando a radiometria média do órgão representado por v_m da seguinte maneira:

$$\hat{\mu_1}(v_m) = \frac{1}{\sum_{k=1}^{N} \sum_{s_i \subset s_{k_l}} |s_i|} \sum_{k=1}^{N} \sum_{(x,y,z) \in s_i, s_i \subset s_{k_l}} I_k(x,y,z) \tag{4.7}$$

e

$$\frac{1}{\sum_{k=1}^{N} \sum_{s_i \subset s_{k_l}} |s_i|} \sum_{k=1}^{N} \sum_{(x,y,z) \in s_i, s_i \subset s_{k_l}} (I_k(x,y,z) - \hat{\mu_1}(v_m))^2. \tag{4.8}$$

$\hat{\mu_1}(v_m)$ é a média amostral da radiometria média, enquanto $\hat{\sigma_1}^2(v_m)$ é sua variância.

TAMANHO. Estimamos os parâmetros da população Normal associada à variável aleatória (X_2) representando o tamanho do órgão representado por v_m da seguinte maneira:

$$\hat{\mu_2}(v_m) = \frac{1}{N} \sum_{k=1}^{N} \sum_{s_i \subset s_{k_l}} |s_i| \tag{4.9}$$

e

$$\hat{\sigma_2}^2(v_m) = \frac{1}{N} \sum_{k=1}^{N} ((\sum_{s_i \subset s_{k_l}} |s_i|) - \hat{\mu_2}(v_m))^2. \tag{4.10}$$

$\hat{\mu_2}(v_m)$ é a média amostral do tamanho de um órgão, enquanto $\hat{\sigma_2}^2(v_m)$ é sua variância.

CENTRÓIDE. Estimamos os parâmetros das populações normais associadas às variáveis aleatórias "x" (X_3^x), "y" (X_3^y), e "z" (X_3^z) representando as coordenadas do centróide do órgão representado por v_m da seguinte maneira:

$$\hat{\mu_3^x}(v_m) = \frac{1}{\sum_{k=1}^{N} \sum_{s_i \subset s_{k_l}} |s_i|} \sum_{k=1}^{N} \sum_{(x,y,z) \in s_i, s_i \subset s_{k_l}} x \tag{4.11}$$

e

$$\hat{\sigma_3^x}^2(v_m) = \frac{1}{\sum_{k=1}^{N} \sum_{s_i \subset s_{k_l}} |s_i|} \sum_{k=1}^{N} \sum_{(x,y,z) \in s_i, s_i \subset s_{k_l}} (x - \hat{\mu_1}(v_m))^2. \tag{4.12}$$

$\hat{\mu_3^x}(v_m)$ é a média amostral da coordenada "x" do centróide, enquanto $\hat{\sigma_3^x}^2(v_m)$ é sua variância. O cálculo dos parâmetros das distribuições associadas às demais coordenadas é feito analogamente.

Agora, considere dois órgãos l_1 e l_2 representados por G_m e relacionados segundo a topologia desse grafo. Considere que esses órgãos são representados por $v_m^i \in V_m$ e $v_m^j \in V_m$ e que os parâmetros estatísticos de suas relações são armazenados em e_m^{ij}. Considere também cada par de nós correspondentes $v_k^i \in V_{obs}^k$ e $v_k^j \in V_{obs}^k$, $k = 1, \ldots, N$, e suas respectivas observações dos atributos relacionais armazenadas em e_k^{ij}. Então estimamos os parâmetros que regem os atributos relacionais desse par de órgãos conforme a descrição a seguir.

CONTRASTE. Estimamos os parâmetros para a população Normal associada à variável aleatória (Y_1) relativa ao contraste entre um par de órgãos representados pelas extremidades de um arco e_m^{ij} da seguinte maneira:

$$\hat{\mu_1^{rel}}(e_m^{ij}) = \frac{1}{N} \sum_{k=1}^{N} y_1(e_k^{ij}) \tag{4.13}$$

e

$$\hat{\sigma_1^{rel}}^2(e_m^{ij}) = \frac{1}{N} \sum_{k=1}^{N} (y_1(e_k^{ij}) - \hat{\mu_1^{rel}}(e_m^{ij}))^2. \tag{4.14}$$

$\hat{\mu_1^{rel}}(e_m^{ij})$ é a média amostral do contraste, enquanto $\hat{\sigma_1^{rel}}^2(e_m^{ij})$ é sua variância.

PROPORÇÃO. Estimamos os parâmetros para a população Normal associada à variável aleatória (Y_2) relativa à proporção entre um par de órgãos representados pelas extremidades de um arco e_m^{ij} da seguinte maneira:

$$\hat{\mu_2}^{rel}(e_m^{ij}) = \frac{1}{N} \sum_{k=1}^{N} y_2(e_k^{ij}) \tag{4.15}$$

e

$$\sigma_2^{\hat{rel}^2}(e_m^{ij}) = \frac{1}{N}\sum_{k=1}^{N}(y_2(e_k^{ij}) - \mu_2^{\hat{rel}}(e_m^{ij}))^2. \tag{4.16}$$

$\mu_2^{\hat{rel}}(e_m^{ij})$ é a média amostral da proporção, enquanto $\sigma_2^{\hat{rel}^2}(e_m^{ij})$ é sua variância.

DISTÂNCIA EUCLIDIANA. Estimamos os parâmetros para a população Normal associada à variável aleatória (Y_3) relativa à distância Euclidiana entre um par de órgãos representados pelas extremidades de um arco e_m^{ij} da seguinte maneira:

$$\mu_3^{\hat{rel}}(e_m^{ij}) = \frac{1}{N}\sum_{k=1}^{n} y_3(e_k^{ij}) \tag{4.17}$$

e

$$\sigma_3^{\hat{rel}^2}(e_m^{ij}) = \frac{1}{N}\sum_{k=1}^{n}(y_3(e_k^{ij}) - \mu_3^{\hat{rel}}(e_m^{ij}))^2. \tag{4.18}$$

$\mu_3^{\hat{rel}}(e_m^{ij})$ é a média amostral da distância euclidiana, enquanto $\sigma_3^{\hat{rel}^2}(e_m^{ij})$ é sua variância.

Aspectos computacionais

Para estimar os parâmetros das variáveis aleatórias representadas por G_m, é necessário acessar todos os dados disponíveis para cada órgão no conjunto de imagens de treinamento. A simples adição de muitos termos pode levar a erros de aritmética computacional relacionados ao limite de armazenamento de valores numa variável de ponto flutuante para estimar a média ou a variância de uma variável aleatória. Portanto, para evitar um possível *overflow* da variável, utilizamos a técnica de estatísticas consecutivas (do inglês, *running statistics*), que consiste numa manipulação das equações para cálculo da média e da variância amostrais, tal que, a cada passo, os valores armazenados são limitados dentro do intervalo de valores que a variável aleatória correspondente pode assumir.

Cada média μ, associada a uma variável aleatória, é calculada da seguinte maneira:

$$\mu = \mu' + \frac{(v - \mu')}{n}, \tag{4.19}$$

e a respectiva variância é medida através do seguinte cálculo:

$$\sigma^2 = \frac{(acc + ((v - \mu') \times (v - \mu)))}{n - 1}, \tag{4.20}$$

sendo que

$$acc = acc + ((v - \mu') * (v - \mu)). \tag{4.21}$$

Portanto, para cada atributo de órgão ou relacional, μ é a média atualizada e estimada para a respectiva variável aleatória, de acordo com as $n \leqslant N$ amostras disponíveis até aquele instante, μ' é a média estimada antes da amostra atual tornar-se disponível, acc é uma variável acumuladora que auxilia a atualização da variância σ^2, e v é o valor medido para a nova amostra disponível. Dessa forma, a cada nova amostra, temos os valores atualizados dos momentos estatísticos para o conjunto de dados já avaliados. Quando todas as amostras tiverem sido disponibilizadas, teremos os valores finais estimados para μ_i e σ_i. O pseudo-código dado pela Figura 25 apresenta uma classe de métodos que processam esses valores para uma dada variável aleatória.

Os valores values μ', acc, e $namostras$ começam com o valor 0.0, e são atualizados conforme novas amostras são consecutivamente disponibilizadas. Então, para determinar a média, a variância, e o desvio padrão, relativos ao conjunto de amostras já processadas, basta chamar os métodos CALCMEDIA, CALCVARIANCIA, CALCDESVPADRAO.

4.4.3 *Segmentação probabilístico-estrutural dos órgãos abdominais*

De posse de um grafo estatístico-relacional expresso por G_m, o objetivo é utilizá-lo para avaliar a segmentação dos órgãos modelados que estejam presentes numa imagem RM abdominal P.

Cada rotulação de P origina um grafo g_{obs} distinto, que pode ser interpretado como uma potencial observação de G_m. Intuitivamente, podemos imaginar que segmentações melhores, segundo o modelo, apresentarão probabilidades maiores do que segmentações de qualidade inferior. Para testar essa hipótese, calculamos $P(G_m = g_{obs})$ para diferentes g_{obs}, de modo que essa probabilidade seja utilizada como uma função-objetivo num processo de otimização para obtenção de um reconhecimento satisfatório dos órgãos em P.

Como discutido na seção 3.3.3, a partir da fórmula genérica dada pela equação 3.3, derivam diversas simplificações dessa probabilidade em termos de dependência condicional, ou de independência, entre as variáveis aleatórias, mas sempre considerando as propriedades dos órgãos e as relações existentes entre eles. A seguir, analisaremos a hipótese de independência para tratar as imagens de ressonância magnética (RM) do abdome.

```
RunningStats::inicializa()
1  namostras = 0
2  μ = 0.0
3  acc = 0.0

RunningStats::novaAmostra(v)
1  // Uma nova amostra de valor v está disponível.
2  namostras = namostras + 1
3  // Se esta é a primeira amostra, a média é igual ao seu valor.
4  // Nesse caso, a variância é indefinida (adotamos o valor 0.0 por convenção).
5  if namostras < 2
6      μ = v
7  else
8      μ' = μ
9      μ = μ' + ((v − μ')/namostras)
10     acc = acc + ((v − μ') * (v − μ))

RunningStats::calcMedia()
1  if namostras < 1
2      return 0.0
3  return μ

RunningStats::calcVariancia()
1  if namostras < 2
2      return 0.0
3  return acc/(namostras − 1))

RunningStats::calcDesvPadrao()
1  if namostras < 2
2      return 0.0
3  return √(acc/(namostras − 1))
```

Figura 25: Pseudo-código para obter as estatísticas consecutivas para um conjunto de amostras.

Probabilidade da solução e a hipótese de independência mútua

O interesse particular na condição de independência mútua provém de sua facilidade de cálculo, já que todos os termos são oriundos de probabilidades independentes. Contudo, por esse mesmo fato, certas dependências entre os órgãos podem não ser consideradas diretamente na modelagem probabilística. Ainda assim, essas dependências são introduzidas pelo modelo através das características relacionais associadas aos arcos.

Além da hipótese de independência entre os órgãos expressa pela Equação 3.4, supomos que as variáveis aleatórias referentes aos atributos de um órgão ou relação são mutuamente independentes. Portanto, cada termo da equação é escrito como:

$$\begin{aligned} P(\overrightarrow{X}(v^i_m) = \overrightarrow{x}(v^i)) &= P(X_1(v^i_m) = x_1(v^i))P(X_2(v^i_m) = x_2(v^i)) \\ &\quad P(X^x_3(v^i_m) = x^x_3(v^i))P(X^y_3(v^i_m) = x^y_3(v^i)) \\ &\quad P(X^z_3(v^i_m) = x^z_3(v^i)), \end{aligned} \tag{4.22}$$

ou

$$\begin{aligned} P(\overrightarrow{Y}(e^j_m) = \overrightarrow{y}(e^j)) &= P(Y_1(e^j_m) = y_1(e^j))P(Y_2(e^j_m) = y_2(e^j)) \\ &\quad P(Y_3(e^j_m) = y_3(e^j)). \end{aligned} \tag{4.23}$$

Como cada uma dessas variáveis aleatórias segue uma distribuição Normal por hipótese, temos, para cada atributo de aparência dado por X_p, $p = 1, 2, 3$, associado a um órgão representado por v^i_m em G_m, e por v^i em g_{obs}:

$$P(X_p(v^i_m) = x_p(v^i)) = \frac{1}{\sqrt{2\pi(\sigma_p(v^i_m))^2}} \exp(-(x_p(v^i))^2/2(\sigma_p(v^i_m))^2). \tag{4.24}$$

De modo similar, para cada relação existente entre dois órgãos, associada à variável aleatória Y_q, $q = 1, 2, 3$, e modelada pelo arco e^j_m em G_m, e por e^j em g_{obs}, temos:

$$P(Y_q(e^j_m) = y_q(e^j)) = \frac{1}{\sqrt{2\pi(\sigma_q(e^j_m))^2}} \exp(-(y_q(e^j))^2/2(\sigma_q(e^j_m))^2). \tag{4.25}$$

Nesta aplicação, a hipótese de que existe independência entre as variáveis é razoável, já que as propriedades de um órgão não são influenciadas umas pelas outras em geral (por ex., o volume de um órgão

não depende de sua radiometria, nem de sua posição). Além disso, as relações entre órgãos distantes são menos significativas (embora úteis) do que aquelas partilhadas entre órgãos vizinhos. Por exemplo, o estômago pode influenciar o volume e posicionamento de órgãos ao seu redor, caso esteja cheio ou vazio, provocando deslocamentos. Contudo, ele não influencia o rim direito, órgão um pouco mais distante, com o qual não tem contato direto.

Na Seção 4.5, discutiremos os resultados experimentais da adoção dessa hipótese no reconhecimento dos órgãos.

ASPECTOS COMPUTACIONAIS E IMPLEMENTAÇÃO. As probabilidades de cada variável aleatória, calculadas nas equações 4.24 e 4.25, podem resultar em valores muito pequenos e muito próximos de zero. Ao multiplicá-las, é necessário considerar a possibilidade de ocorrência de *underflow*. Num sistema Linux Ubuntu, tal como o que usamos em nossos experimentos, um *underflow* ocorre quando uma variável tenta armazenar um valor abaixo de $2.22507e-308$. Para tratar esse problema, calculamos o $\log(P(G_m = g_{obs}))$ ao invés da probabilidade diretamente, o que nos leva ao cálculo de uma soma de logaritmos de probabilidades, ao invés da multiplicação dos termos individualmente:

$$\begin{aligned} \log(P(G_m = g_{obs})) \quad = \quad & \sum_{i=1,\ldots,|V_m|} \log(P(\overrightarrow{X}(v_m^i) = \overrightarrow{x}(v^i))) \\ & \sum_{j=1,\ldots,|E_m|} \log(P(\overrightarrow{Y}(e_m^j) = \overrightarrow{y}(e^j))). \end{aligned} \tag{4.26}$$

Portanto, ao invés de multiplicarmos diversos valores entre 0 e 1, transformamos esse produto de probabilidades numa soma de logaritmos, sem que haja comprometimento da ordem relativa entre os valores. Contudo, os logaritmos também são suscetíveis a *underflows*, já que $-\infty \leqslant \log x \leqslant 0$. Portanto, é preciso cuidar para que essa soma não resulte num valor inferior ao menor valor negativo representável para um dado tipo de dado numérico (neste caso, `double`).

Adotamos a política de substituir um logaritmo que resulte em $-\infty$ por um valor constante e suficientemente negativo. Embora haja um impacto sobre o valor da soma (probabilidade) final, esses valores "singulares" ocorrem principalmente quando a solução local (Equações 4.24 e 4.25) é muito ruim em relação ao modelo. Portanto, é menos crítico que essas soluções sejam atribuídas a um valor negativo suficientemente pequeno, ou ao verdadeiro valor $-\infty$, já que uma soma com muitas soluções locais ruins terá mais fatores iguais a essas constantes negativas. O importante é que o método ainda seja capaz de detectar quando uma solução é ruim e de penalizá-la adequadamente. A adição dessas contantes negativas permite, de fato, sinalizar a ocorrência de uma solução ruim, além de fornecer um va-

lor numérico global para $P(G_m = g_{obs})$ que possa ser comparado a outros valores obtidos para outras segmentações.

4.5 RESULTADOS EXPERIMENTAIS

O protocolo experimental descrito a seguir teve por objetivo avaliar a eficácia do modelo estatístico-relacional do abdome (Seção 4.4.2), e da hipótese de independência aplicada à probabilidade $P(G_m = g_{obs})$ (Equações 4.22 e 4.26), em discriminar a qualidade de diferentes segmentações possíveis dos órgãos abdominais de interesse sobre uma imagem de entrada. Para esse fim, o conjunto de experimentos consistiu nos seguintes passos:

1. dado um conjunto de treinamento de imagens RM do abdome, segmentamos manualmente os órgãos abdominais de interesse sobre cada uma dessas imagens, associando um rótulo único aos voxels que pertencem a um dado órgão, resultando em máscaras com rótulos anotados para cada imagem de treinamento (Seção 4.5.1);

2. calculamos um grafo estatístico-relacional a partir desse conjunto de treinamento já segmentado manualmente e anotado (Seção 4.5.1);

3. para cada imagem de treinamento segmentada, calculamos seu respectivo grafo relacional com atributos, que corresponde a um grafo de observação obtido de uma segmentação ideal (*ground truth*) dessa imagem (Seção 4.5.2);

4. modificamos ligeiramente cada segmentação manual original, para que correspondessem a uma segmentação imperfeita, mas pouco distante da original; então calculamos o seu respectivo grafo de observação (Seção 4.5.3). A modificação realizada consistiu simplesmente na dilatação da região rotulada como fígado na segmentação ideal por um elemento estruturante 3D (18-conectividade);

5. trocamos os rótulos das regiões segmentadas como rim direito e rim esquerdo em cada segmentação ideal, e calculamos seus respectivos grafos de observação (Seção 4.5.3). Essa modificação teve por objetivo simular um erro mais dramático de segmentação em relação à segmentação ideal;

6. trocamos, aleatoriamente, os rótulos de uma porcentagem do total de voxels de uma segmentação ideal, produzindo uma nova segmentação e seu respectivo grafo de observação (Seção 4.5.3). A porcentagem de erros introduzidos nas segmentações varia sistematicamente entre 1% e 10% do total de voxels, com aumento de 1% na taxa de erros gerados entre uma segmentação

e outra, e de 10% a 100%, com aumento de 10% na taxa de erros gerados entre uma segmentação e outra;

7. calculamos a probabilidade de cada grafo de observação gerado nos passos anteriores em relação ao GER modelo calculado no segundo passo.

8. comparamos as probabilidades obtidas, para avaliar o comportamento do método quanto à sua capacidade de discriminar entre soluções distintas (Seção 4.5.4).

Esses passos serão explicados em detalhe a seguir.

4.5.1 *Grafo estatístico-relacional*

O conjunto de treinamento.

O conjunto de treinamento consiste nas $N = 4$ imagens RM ilustradas na Figura 26, cada uma correspondendo a um paciente distinto identificado como P_1, P_2, P_3, ou P_4, e todas obtidas por protocolo do tipo T1 de um mesmo equipamento. Essas imagens foram escolhidas por apresentarem melhor qualidade do que outras imagens disponíveis em termos de visibilidade dos órgãos.

Embora esse número de amostras de treinamento seja pequeno para estimar as estatísticas desejadas de modo robusto, ele é justificado pela dificuldade em se obter muitas rotulações manuais a partir das imagens abdominais de ressonância magnética, já que, para cada uma, é preciso rotular os órgãos de interesse no volume inteiro, como explicaremos a seguir.

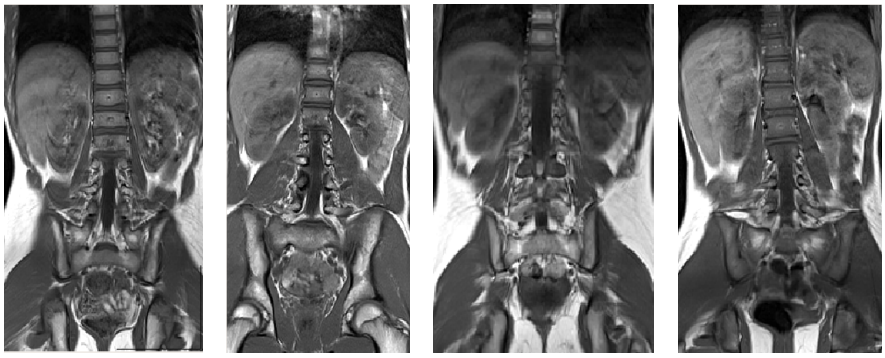


Figura 26: Imagens RM selecionadas para compor o conjunto de treinamento que originou o grafo estatístico-relacional aplicado aos experimentos.

Em cada imagem de treinamento, segmentamos manualmente os órgãos denotados pelo conjunto L descrito na Seção 4.3, ou seja, fígado, rins direito e esquerdo, estômago, e coluna, além do elemento genérico dito "fundo". Um único rótulo identificador foi associado a

Órgão	Rótulo Identificador	Cor (RGB)
Fígado	51	(0,0,255) (azul)
Rim direito	52	(118,238,198) (ocre)
Rim esquerdo	53	(255,69,0) (verde)
Estômago	54	(255,62,150) (rosa)
Coluna vertebral	50	(193,205,205) (cinza)
Fundo	0	(0,0,0) (preto)

Tabela 2: Correspondência entre órgãos, rótulos identificadores, e cores para identificação visual.

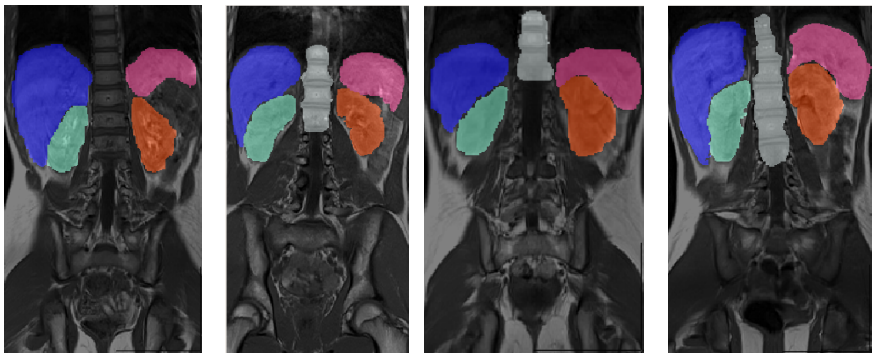


Figura 27: Imagens de treinamento após a segmentação manual dos órgãos abdominais: fígado (azul), estômago (rosa), rim direito (ocre), rim esquerdo (verde), e coluna vertebral (cinza).

cada órgão, e tal rótulo também corresponde visualmente a uma cor característica. A Tabela 2 indica essa associação entre órgãos, rótulos e cores. Essa mesma convenção de rótulos e cores será adotada para todas as imagens segmentadas, bem como para a etapa de segmentação dos órgãos.

Todas as segmentações manuais das imagens fornecidas pelo Hospital Saint Vincent de Paul foram realizadas através do software *3D Slicer* e consistiram na criação de uma imagem máscara para cada imagem de treinamento, na qual as regiões correspondentes aos órgãos modelados foram marcadas com seus respectivos rótulos identificadores, e os demais voxels foram marcados com o identificador para o fundo. Essas máscaras foram validadas pela radiologista Céline Falip, da equipe de radiologia pediátrica do Hospital Saint Vincent de Paul. A Figura 27 exibe alguns cortes correspondentes às máscaras geradas para cada imagem. Também nos referiremos a essas segmentações manuais como *segmentações ideais*, pois reproduzem a segmentação que um especialista realizaria sobre esses dados RM.

Grafo estatístico-relacional: resultados da etapa de treinamento

O grafo estatístico-relacional G_m foi construído segundo as imagens originais dos pacientes P_1, P_2, P_3, e P_4 (Figura 26) e suas respectivas máscaras rotuladas (Figura 27). Após a etapa de treinamento, cada nó ou arco de G_m armazena os parâmetros estimados para cada variável aleatória associada aos atributos de aparência ou relacionais dos órgãos, os quais são apresentados nas Tabelas 3 e 4. O cálculo desses parâmetros foi realizado conforme as explicações dadas na Seção 4.4.2, através do estimador de máxima verossimilhança.

Tabela 3: Estatísticas (média e desvio-padrão) relativas às variáveis aleatórias (v.a.) que representam os atributos de aparência dos órgãos que compõem o GER G_m. Esses parâmetros foram estimados com base nas segmentações manuais das imagens do conjunto de treinamento.

Órgão	**Estatísticas dos atributos de aparência (μ, σ) por paciente/imagem**				
	Radiometria (X_1)	Tamanho (X_2)	Centróide "x" (X_3^x)	Centróide "y" (X_3^y)	Centróide "z" (X_3^z)
Fígado	$\mathcal{N}(372.79, 80.08)$	$\mathcal{N}(51483.75, 28632.04)$	$\mathcal{N}(71, 35)$	$\mathcal{N}(46, 27)$	$\mathcal{N}(5, 3)$
Estômago	$\mathcal{N}(326.28, 104.58)$	$\mathcal{N}(25310.25, 15710.9)$	$\mathcal{N}(67, 24)$	$\mathcal{N}(134, 19)$	$\mathcal{N}(6, 3)$
Rim Dir.	$\mathcal{N}(316.81, 87.67)$	$\mathcal{N}(6943.75, 3517.66)$	$\mathcal{N}(110, 21)$	$\mathcal{N}(46, 12)$	$\mathcal{N}(9, 1)$
Rim Esq.	$\mathcal{N}(314.83, 94.02)$	$\mathcal{N}(8736, 2925.58)$	$\mathcal{N}(100, 22)$	$\mathcal{N}(129, 14)$	$\mathcal{N}(9, 2)$
Coluna	$\mathcal{N}(290.28, 109.58)$	$\mathcal{N}(18455.33, 5153.47)$	$\mathcal{N}(117, 55)$	$\mathcal{N}(86, 10)$	$\mathcal{N}(6, 2)$
Fundo	$\mathcal{N}(311.27, 203.06)$	$\mathcal{N}(702372.25, 291661.65)$	$\mathcal{N}(168, 93)$	$\mathcal{N}(89, 52)$	$\mathcal{N}(7, 5)$

4.5.2 *Segmentações ideais*

As Tabelas 5 e 6 apresentam os valores observados para cada variável aleatória associada a um atributo de aparência de um órgão, ou a um atributo relacional entre órgãos, para cada imagem de treinamento (Figura 26) correspondente a um dos pacientes P_1, P_2, P_3, ou P_4. Esses valores correspondem àqueles armazenados nos respectivos grafos de observação gerados para cada imagem e sua respectiva máscara. Tomamos esses grafos de observação como referenciais, ou seja, representam as segmentações ideais para as respectivas imagens.

É importante ressaltar que, nas imagens do paciente P_1, a coluna vertebral não foi rotulada como tal, tendo sido considerada como fundo. Isso implica que as regiões pertencentes à coluna vertebral, nessa imagem, foram consideradas nas estatísticas de fundo na etapa de treinamento, e que o grafo de observação oriundo dessa segmentação de P_1 não contém um nó para a coluna vertebral. Desconsiderar esse órgão permitiu avaliar o comportamento da metodologia proba-

Tabela 4: Estatísticas (média e desvio-padrão) relativas às variáveis aleatórias (v.a.) associadas aos atributos relacionais entre órgãos representados pelo GER G_m. Esses parâmetros foram estimados com base nas segmentações manuais das imagens do conjunto de treinamento.

Atributos Relacionais	**Estatísticas dos atributos relacionais (μ, σ) por paciente/imagem**		
	Contraste (Y_1)	Proporção (Y_2)	Distância (Y_3)
Fígado,Estômago	$\mathcal{N}(0.07, 0.03)$	$\mathcal{N}(0.47, 0.14)$	$\mathcal{N}(90.46, 6.59)$
Fígado,Rim Dir.	$\mathcal{N}(0.07, 0.02)$	$\mathcal{N}(0.15, 0.06)$	$\mathcal{N}(39.35, 6.59)$
Fígado,Coluna	$\mathcal{N}(0.11, 0.09)$	$\mathcal{N}(0.32, 0.07)$	$\mathcal{N}(1.08, 13.07)$
Fígado,Fundo	$\mathcal{N}(0.11, 0.06)$	$\mathcal{N}(0.08, 0.04)$	$\mathcal{N}(103.53, 13.42)$
Rim Dir.,Rim Esq.	$\mathcal{N}(0.02, 0.02)$	$\mathcal{N}(0.78, 0.16)$	$\mathcal{N}(81.17, 7.76)$
Rim Dir.,Fundo	$\mathcal{N}(0.07, 0.06)$	$\mathcal{N}(0.01, 0.002)$	$\mathcal{N}(68.63, 16.73)$
Rim Esq.,Estômago	$\mathcal{N}(0.03, 0.02)$	$\mathcal{N}(0.48, 0.34)$	$\mathcal{N}(36.05, 10.44)$
Rim Esq.,Fundo	$\mathcal{N}(0.07, 0.06)$	$\mathcal{N}(0.01, 0.004)$	$\mathcal{N}(76.51, 13.57)$
Estômago,Coluna	$\mathcal{N}(0.09, 0.08)$	$\mathcal{N}(0.6, 0.06)$	$\mathcal{N}(69.06, 12.12)$
Estômago,Fundo	$\mathcal{N}(0.09, 0.08)$	$\mathcal{N}(0.04, 0.02)$	$\mathcal{N}(109.67, 14.17)$
Coluna,Fundo	$\mathcal{N}(0.04, 0.05)$	$\mathcal{N}(0.03, 0.006)$	$\mathcal{N}(51.32, 0.54)$

blística quando da ausência, na imagem, de um órgão presente no modelo, conforme mostraremos na Seção 4.5.4.

Tabela 5: Observações medidas a partir de cada imagem original e de sua respectiva rotulação ideal, relativas às características de aparência de cada órgão: x_1 (radiometria média), x_2 (tamanho), x_3^x (coordenada "x" do centróide), x_3^y (coordenada "y" do centróide), e x_3^z (coordenada "z" do centróide).

Atributos de aparência		**Observações dos atributos de aparência / paciente**			
		P_1	P_2	P_3	P_4
Fígado	radiometria média	396.93	341.18	369.57	403.48
	tamanho	22182	81506	32426	69821
	centróide	(79,38,8)	(86,50,5)	(55,37,4)	(59,49,4)
Estômago	radiometria média	331.3	295.54	307.8	375.48
	tamanho	6177	42131	19803	33130
	centróide	(63,137,8)	(86,139,6)	(52,123,4)	(52,135,5)
Rim Dir.	radiometria média	356.63	281.94	330.52	344.05
	tamanho	5141	12187	4743	5704
	centróide	(124,56,9)	(118,42,9)	(89,41,6)	(99,51,8)
Rim Esq.	radiometria média	322.27	292.65	321.82	334.75
	tamanho	6034	12600	6978	9332
	centróide	(114,137,9)	(114,134,10)	(79,113,7)	(87,128,8)
Coluna	radiometria média	0	277.94	351.11	263.81
	tamanho	0	23268	13018	19080
	centróide	0	(134,88,8)	(82,76,5)	(119,91,6)
Fundo	radiometria média	333.6	275.77	419.03	271.96
	tamanho	704014	1105648	427159	572668
	centróide	(161,91,5)	(185,192,9)	(133,78,6)	(171,90,6)

Tabela 6: Observações medidas a partir de cada imagem original e de sua respectiva rotulação ideal, relativas às características relacionais entre pares de órgãos: y_1 (contraste), y_2 (proporção), e y_3 (distância).

Atributos relacionais		**Observações dos atributos relacionais / paciente**			
		P_1	P_2	P_3	P_4
Fígado,Estômago	contraste	0.09	0.072	0.09	0.04
	proporção	0.28	0.52	0.61	0.47
	distância	100.15	89.1	86.39	86.21
Fígado,Rim Dir.	contraste	0.05	0.1	0.06	0.08
	proporção	0.23	0.15	0.15	0.08
	distância	48.48	33.15	34.97	40.78
Fígado,Coluna	contraste	0	0.1	0.03	0.21
	proporção	0	0.29	0.4	0.27
	distância	0	61.37	47.87	74.0
Fígado,Fundo	contraste	0.09	0.11	0.06	0.19
	proporção	0.03	0.07	0.08	0.12
	distância	97.44	107.6	88.98	120.11
Rim Dir.,Rim Esq.	contraste	0.05	0.02	0.01	0.01
	proporção	0.85	0.97	0.68	0.61
	distância	81.7	91.65	73.44	77.9
Rim Dir.,Fundo	contraste	0.03	0.01	0.12	0.12
	proporção	0.01	0.01	0.01	0.01
	distância	50.78	83.52	57.96	82.24
Rim Esq.,Estômago	contraste	0.01	0.005	0.02	0.06
	proporção	0.98	0.3	0.35	0.28
	distância	50.89	28.31	29.22	35.78
Rim Esq.,Fundo	contraste	0.02	0.03	0.13	0.1
	proporção	0.01	0.01	0.02	0.02
	distância	66.21	82.96	64.35	92.52
Estômago,Coluna	contraste	0	0.03	0.07	0.18
	proporção	0	0.55	0.66	0.58
	distância	0	70.22	56.41	80.56
Estômago,Fundo	contraste	0.004	0.03	0.15	0.16
	proporção	0.009	0.04	0.05	0.06
	distância	108.35	109.89	92.9	127.54
Coluna,Fundo	contraste	0	0.004	0.09	0.02
	proporção	0	0.02	0.03	0.03
	distância	0	51.13	50.89	51.93

4.5.3 *Variações sobre as segmentações ideais*

Para gerar variações das segmentações ideais, realizamos uma série de modificações a partir das máscaras ideais e suas respectivas imagens originais (Figuras 26 e 27).

Em primeiro lugar, simulamos segmentações ligeiramente distintas das ideais. Já que uma segmentação manual é inerentemente subjetiva, o objetivo dessas alterações foi simular uma segmentação realizada por um outro especialista, com variações mínimas. Para essa finalidade, aplicamos automaticamente um operador de dilatação morfológica apenas ao conjunto de regiões rotuladas como fígado. O elemento estruturante tridimensional de conectividade 18 foi utilizado nos testes. A Figura 28 exibe as máscaras resultantes (rotulações) dessas dilatações, sobrepostas às imagens de treinamento.

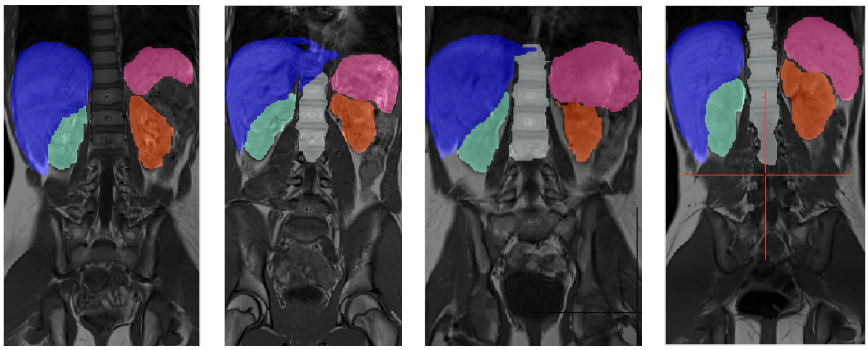


Figura 28: Máscaras obtidas a partir da dilatação dos voxels pertencentes ao fígado (regiões azuis) por um elemento estruturante 3D (18-conectividade), sobrepostas às imagens originais. Essas dilatações foram realizadas para simular segmentações ligeiramente distintas das ideais.

A segunda manipulação dos dados das segmentações ideais teve por objetivo simular segmentações nas quais alguns órgãos foram rotulados de modo completamente incorreto. Para esse fim, invertemos automaticamente os rótulos das regiões segmentadas como rim direito e rim esquerdo. Como essas duas estruturas anatômicas são relativamente similares em termos de características de aparência, elas podem ser mal classificadas. Por outro lado, o fato de estarem em lados opostos do corpo humano e de partilharem relações com órgãos distintos em suas vizinhanças deveria desfavorecer esse tipo de segmentação errônea. Através da troca desses rótulos, poderemos avaliar o comportamento da probabilidade dessas observações, e verificar se o método de fato considera esses aspectos. A Figura 29 apresenta as máscaras resultantes dessa manipulação, sobrepostas às imagens de treinamento.

Uma última série de modificações das rotulações ideais consistiu na troca dos rótulos de voxels aleatórios de maneira incremental. As Figuras 30 e 31 apresentam algumas dessas novas rotulações, sobre-

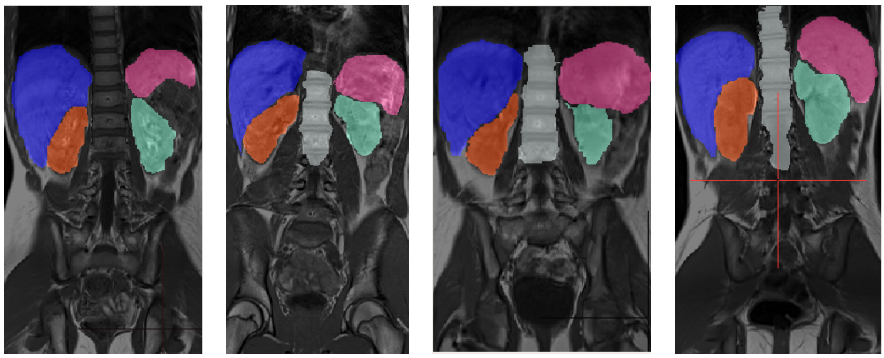


Figura 29: Máscaras obtidas a partir da inversão dos rótulos das regiões pertencentes ao rim direito (ocre) e ao rim esquerdo (verde), sobrepostas às imagens originais. Essas inversões foram realizadas para simular segmentações com órgãos rotulados incorretamente.

postas às imagens originais. A quantidade de rótulos alterados aleatoriamente foi controlada e dividida em dois intervalos: de 1% a 9% do total de voxels, em passos de 1% entre uma rotulação aleatória e a seguinte; depois, de 10% a 100%, com passos de 10%. Em todas as rotulações aleatórias, asseguramos que o novo rótulo atribuído a um dado voxel é distinto do rótulo ideal.

O primeiro intervalo produz rotulações cujas quantidades de erros são relativamente pequenas em relação à respectiva rotulação ideal. Por outro lado, o segundo intervalo produz rotulações cada vez mais incorretas. Esses dois cenários permitiram avaliar o comportamento de $P(G_m = g_{obs})$ e sua sensibilidade aos erros em diferentes rotulações.

4.5.4 *Avaliação das segmentações*

As Tabelas 32-34 apresentam o comportamento da probabilidade $P(G_m = g_{obs})$, calculada segundo a Equação 4.26, para cada GRA g_{obs} extraído de uma imagem resultante de variações das imagens ideais, em relação às informações estatísticas armazenadas no GER G_m, criado segundo as explicações dadas na Seção 4.3.

A Tabela 32 apresenta os valores dos logaritmos das probabilidades obtidos para as rotulações ideais realizadas manualmente para cada imagem original associada a um certo paciente (P_1, P_2, P_3, P_4), bem como para as rotulações em que o fígado foi dilatado, ou nas quais os rótulos dos rins esquerdo e direito foram trocados. Em todos os testes, a probabilidade se comportou da maneira esperada intuitivamente: para cada paciente, a probabilidade da rotulação ideal superou as probabilidades de todas as suas variações. Em relação às variações, quanto mais expressivos os erros, menor a probabilidade correspondente na grande parte dos casos. Portanto, as rotulações geradas pela

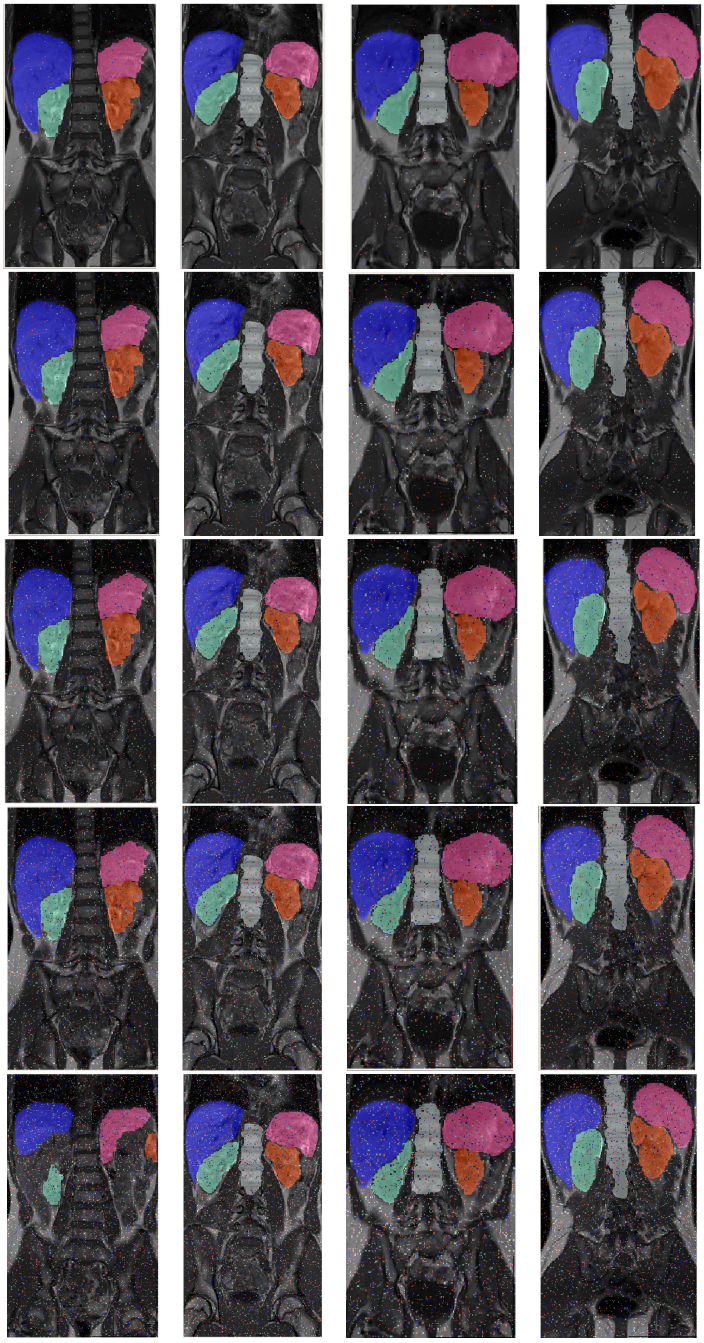


Figura 30: Máscaras obtidas através da troca incremental de uma porcentagem dos rótulos que compõem a segmentação ideal para cada paciente. Da esq. para a dir.: as colunas correspondem às imagens P_1, P_2, P_3, e P_4, respectivamente, com sobreposição de suas máscaras referentes às segmentações com erros aleatórios. De cima para baixo, cada linha apresenta os resultados para uma dada taxa de erros, correspondente a 1%, 3%, 5%, 7%, e 9% do total de voxels da imagem.

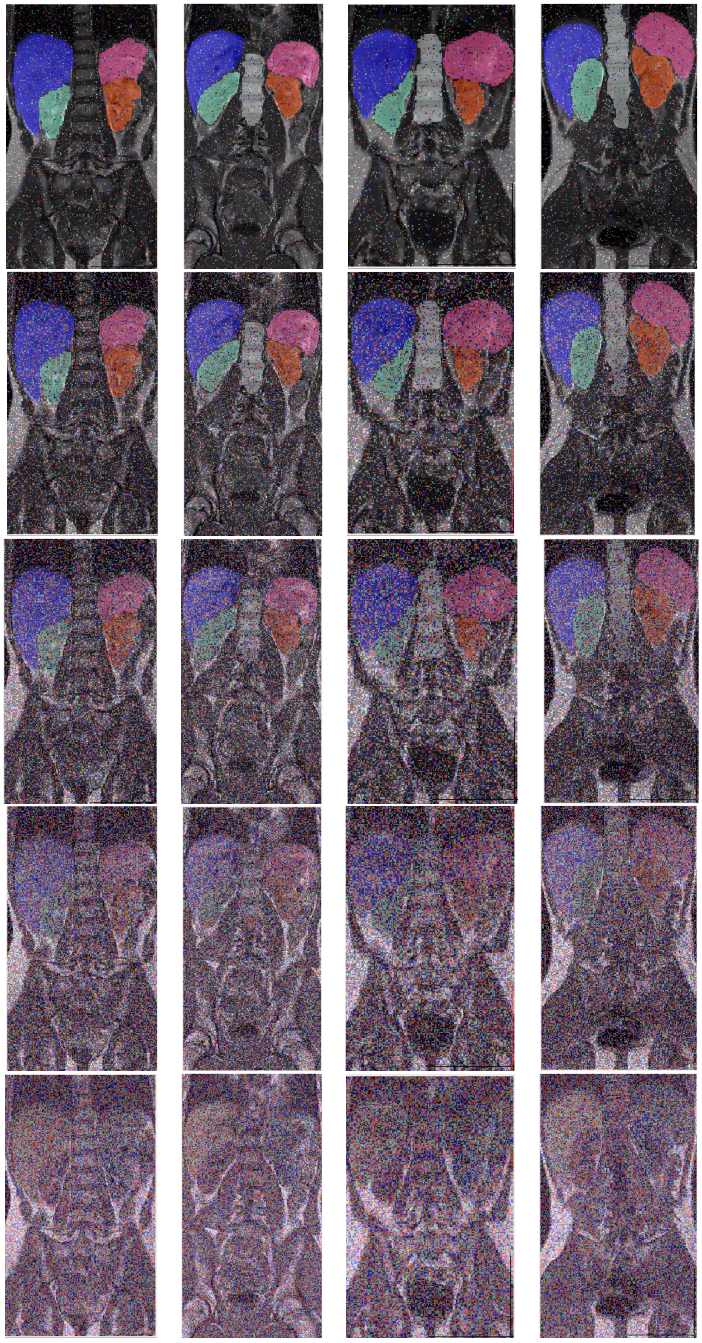


Figura 31: Máscaras obtidas através da troca incremental de uma porcentagem dos rótulos que compõem a segmentação ideal para cada paciente. Da esq. para a dir.: as colunas correspondem às imagens P_1, P_2, P_3, e P_4, respectivamente, com sobreposição de suas máscaras referentes às segmentações com erros aleatórios. De cima para baixo, cada linha apresenta os resultados para uma dada taxa de erros, correspondente a 10%, 30%, 50%, 70%, e 90% do total de voxels da imagem.

dilatação do fígado têm maior probabilidade do que aquelas obtidas a partir da troca dos rótulos de rins.

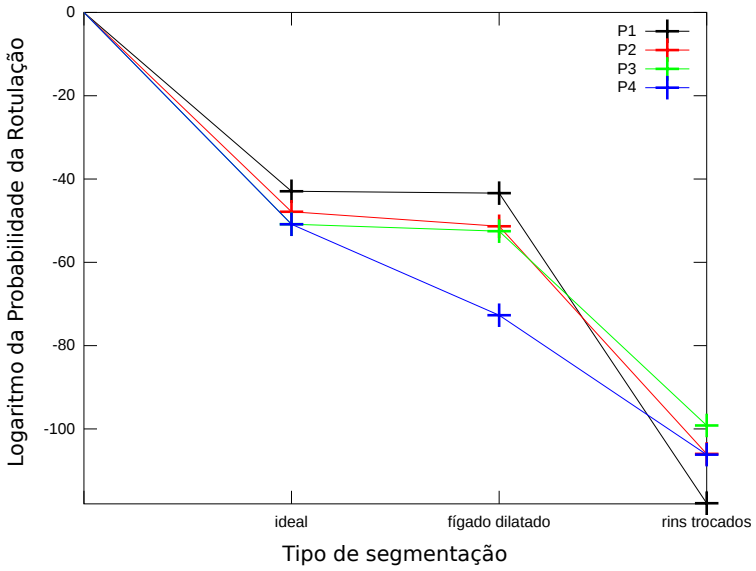


Figura 32: Logaritmos das probabilidades segundo os dados estatísticos contidos no GER G_m (Seção 4.5.1), calculados para: rotulação ideal; ligeira modificação da rotulação ideal, resultante da dilatação da máscara do fígado; modificação da rotulação ideal, resultante da troca dos rótulos dos rins direito e esquerdo. Cada linha representa os resultados para cada paciente e suas respectivas imagens resultantes.

As Tabelas 33 e 34 apresentam os logaritmos das probabilidades obtidos para as rotulações resultantes de trocas aleatórias de rótulos dos órgãos nas máscaras ideais originais. Na primeira tabela, os resultados exibidos correspondem ao intervalo de erros aleatórios de 1% a 9% do total de voxels, enquanto a segunda tabela apresenta os resultados para o intervalo de 10% a 100% de erros aleatórios.

O comportamento dos logaritmos das probabilidades também se repetiu em relação às rotulações geradas a partir de erros aleatórios adicionados ao *ground truth*. Contudo, podemos notar alguns fenômenos distintos em relação a esse comportamento, sobretudo quando a quantidade de erros ultrapassa 80% do total de voxels. Esse fato pode indicar que o método tem menor capacidade para discriminar a melhor solução dentre duas rotulações muito ruins. Esse mesmo fenômeno foi observado para as primeiras rotulações com erros aleatórios (1% e 2%) geradas para P_1. Como mencionado previamente, a rotulação ideal de P_1 não inclui a coluna vertebral, que acaba sendo considerada como fundo. Através da troca aleatória de rótulos, poderemos ter voxels e regiões rotuladas como coluna vertebral, mas que são muito diferentes quando comparadas às estatísticas da coluna vertebral contidas em G_m. Portanto, nesses casos, o logaritmo

da probabilidade pode apresentar um comportamento não monotonicamente decrescente em relação a menos erros.

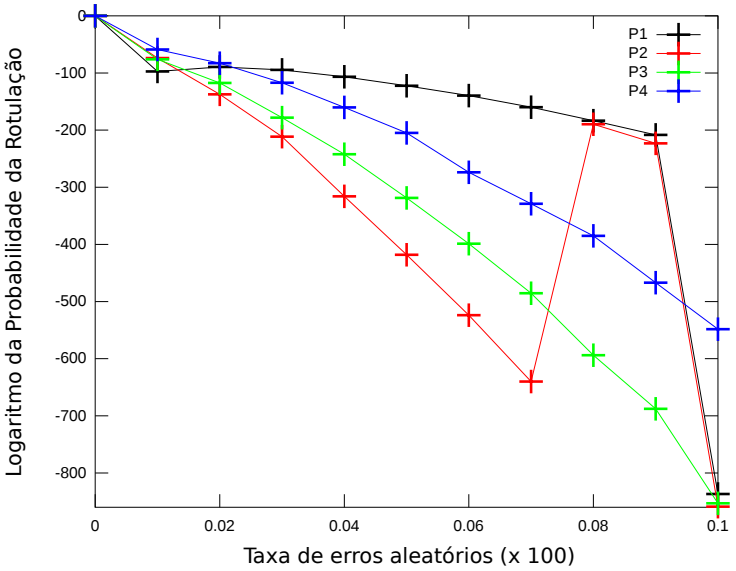


Figura 33: Logaritmo das probabilidades das rotulações modificadas através da troca sistemática de rótulos a partir de cada rotulação ideal. Conforme os erros aumentam, os logaritmos das probabilidades tendem a cair.

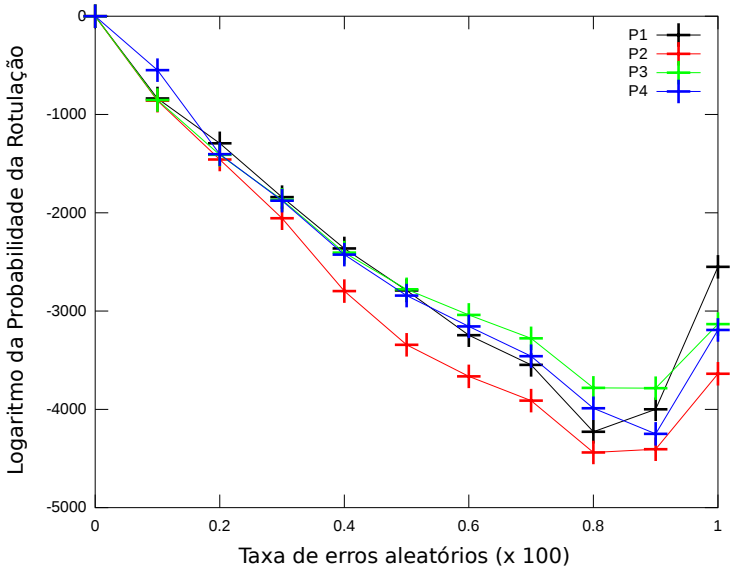


Figura 34: Logaritmo das probabilidades das rotulações modificadas através da troca sistemática de rótulos a partir de cada rotulação ideal. Conforme os erros aumentam, os logaritmos das probabilidades tendem a cair, exceto quando as rotulações obtidas são muito ruins.

Embora o modelo utilizado esteja apoiado em hipóteses fortes, tais como $|V_{obs}| = |V_m|$, e independência total entre as variáveis aleatórias que descrevem os órgãos e suas relações, os resultados práticos mostram que esse modelo simplificado permite avaliar diferentes rotulações e discriminá-las de maneira bastante satisfatória de acordo com os dados de aparência e relacionais disponíveis. Ainda, esses resultados corroboram a hipótese de que rotulações (segmentações) mais próximas da ideal, de acordo com o modelo estatístico, são mais prováveis do que rotulações ruins.

5 CONSIDERAÇÕES FINAIS

Digo: o real não está na saída nem na chegada:
ele se dispõe para a gente é no meio da travessia.

— João Guimarães Rosa, em *Grande Sertão: Veredas*

5.1 RESUMO DAS CONTRIBUIÇÕES

Nesta tese, apresentamos uma metodologia para modelar objetos estruturados, combinando o conhecimento acerca do arranjo estrutural que os caracteriza, com as estatísticas estimadas para características que descrevem a aparência de seus elementos primitivos e as relações entre eles.

Através desse modelo híbrido, o problema de reconhecimento de objetos estruturados em imagens foi formulado sob um enfoque probabilístico, que avalia a pertinência de uma dada rotulação de uma imagem de entrada de acordo com o modelo. Assim, o modelo é interpretado como uma variável aleatória cujas leis são conhecidas, e o reconhecimento derivado de uma rotulação é interpretado como uma potencial observação dessa variável, cuja probabilidade desejamos conhecer. Esse formalismo determina uma função-objetivo, ou energia, que é adotada no processo de otimização do reconhecimento final almejado. Assim, as características estatístico-estruturais são incorporadas, por meio dessa função, ao algoritmo de otimização escolhido.

Essa proposta se insere numa vertente de trabalhos relacionados a reconhecimento de objetos, que pretende aproveitar os conhecimentos provenientes do histórico de estudos em reconhecimento estatístico de padrões, juntamente com os métodos de reconhecimento estrutural de padrões, dotados de um grande poder de representação e sustentados por sólidas teorias, como grafos e autômatos.

O levantamento bibliográfico (Capítulo 2) dos trabalhos do estado-da-arte que discutem essa conjunção de paradigmas aponta para a possibilidade de aplicação de métodos estatísticos em quaisquer etapas de um processo de reconhecimento, desde a representação dos objetos ao reconhecimento final. Nesta tese, buscamos adotar esse hibridismo tanto na modelagem de um objeto estruturado, quanto no seu reconhecimento em imagens.

A seguir, discutiremos as principais contribuições oriundas do trabalho, assim como seus desafios e limitações.

5.1.1 *Modelagem híbrida de objetos estruturados*

Para considerar a complementaridade entre os enfoques estatístico e estrutural para reconhecimento de padrões, propusemos uma representação para um objeto ou sistema estruturado por grafo estatístico-relacional (GER), a qual permitiu integrar, numa única estrutura de dados, as fontes de informação estatística e relacional que o descrevem. Além disso, a definição suficientemente generalista de um GER o torna igualmente adequado para representar objetos ou sistemas estruturados bidimensionais, ou tridimensionais, de maneira uniforme.

As estatísticas que modelam as variáveis aleatórias representantes de atributos de aparência e relacionais de um elemento primitivo foram naturalmente integradas a esse grafo por meio dos vetores de atributos de que um GRA dispõe. Em particular, no caso do GER, esses vetores armazenam os parâmetros das distribuições de probabilidade estimadas para cada cada variável aleatória. Por outro lado, a descrição estrutural e relacional típicas de um objeto estruturado foram integradas no GER através de sua topologia.

Devido a essas características da modelagem híbrida, a criação de um GER compreende os seguintes subproblemas:

- definição dos elementos primitivos que descreverão o objeto estruturado, e determinarão o número de nós do grafo;
- definição das relações entre os elementos primitivos, que determinarão a topologia do grafo;
- escolha das características de aparência que descreverão cada elemento primitivo, e cujos parâmetros serão estimados e armazenados nos vetores de atributos dos nós do GER;
- escolha das características relacionais que descreverão quantitativamente as relações entre elementos primitivos, e cujos parâmetros serão estimados e armazenados nos vetores de atributos dos arcos do GER;
- escolha e disponibilidade de um conjunto de treinamento referente ao objeto estruturado em questão, para realização do aprendizado supervisionado das estatísticas de suas características de aparência e relacionais.

A determinação dos elementos primitivos e de suas relações é transferida para a linguagem de grafos, que são extensivamente adotados no paradigma de reconhecimento estrutural de padrões. Já os três últimos subproblemas estão diretamente relacionados ao paradigma de reconhecimento estatístico.

Em particular, a escolha de características de aparência e relacionais discriminantes pode ser feita de diferentes maneiras, segundo a aplicação de interesse e os dados disponíveis: através da simples

inspeção visual de diferentes amostras de um objeto estruturado, ou através da análise de especificações conhecidas de seus elementos primitivos (como no caso da modelagem de órgãos abdominais), ou, de maneira mais robusta, através dos métodos automáticos de seleção de características [JZ97].

Uma vez determinadas as características, a estimação de parâmetros das distribuições de probabilidade das variáveis aleatórias que as modelam pode ser realizada com apoio de métodos clássicos de aprendizado supervisionado estatístico, como, por exemplo, os estimadores de parâmetros Bayesiano, ou de máxima verossimilhança.

Desses fatos, fica evidente como as técnicas pré-existentes, oriundas de ambos os paradigmas, podem ser reutilizadas na construção do GER, bem como o grande potencial de representação desse grafo.

Para validar a modelagem híbrida experimentalmente, propusemos sua aplicação à descrição de órgãos abdominais em imagens de ressonância magnética, um domínio que se mostrou bastante desafiador e adequado para ser tratado dentro da abordagem estatístico-estrutural proposta devido às suas características intrínsecas.

5.1.2 *Abordagem probabilística para reconhecimento de objetos estruturados*

O segundo aspecto da metodologia proposta consistiu em realizar o reconhecimento em imagens de um objeto estruturado à luz de seu respectivo GER. Dada uma imagem contendo um objeto estruturado, desejamos reconhecer e agrupar seus pixels ou voxels de forma que, ao final, obtenhamos conjuntos de regiões associadas aos elementos primitivos modelados e que estão presentes na imagem. Nesse cenário, o desafio consistiu em utilizar efetivamente os dados estatísticos e as informações relacionais expressas pelo GER para definir essa associação entre o conteúdo da imagem e o modelo.

Dado que a representação digital de uma imagem (informações de baixo nível) não está na mesma escala de interpretação do modelo (conhecimento de alto nível), propusemos uma representação por grafo de observação, no intuito de conectar esses dois níveis de representação e, então, proceder ao reconhecimento baseado em modelo.

Como lidamos com objetos estruturados e um modelo descrito por um grafo, tratamos o problema de reconhecimento sob uma perspectiva probabilística relacionada ao problema de casamento entre grafos. Essa interpretação levou à formulação de uma função-objetivo para comparar o grafo modelo G_m e o grafo de observação g_{obs}, que consiste, na verdade, na probabilidade de se observar uma rotulação da entrada, dado o grafo modelo. O reconhecimento almejado é definido como um problema de otimização dessa probabilidade.

O comportamento dessa função-objetivo foi estudado em situações distintas através de simulações com dados sintéticos, bem como num

quadro experimental prático. Nesse último caso, estudamos o comportamento da função em relação ao reconhecimento de órgãos abdominais em imagens de ressonância magnética, os quais foram modelados através de um GER. Nesses dois cenários, estudamos a hipótese de independência mútua entre os elementos primitivos e as variáveis aleatórias que os descrevem.

Embora essa hipótese desconsidere as dependências que possam existir entre os elementos primitivos do modelo, ela permite estimar, de modo simples, os parâmetros das variáveis aleatórias de aparência e relacionais, facilitando o cálculo da probabilidade de uma rotulação dada por $P(G_m = g_{obs})$.

Os resultados obtidos experimentalmente apontam que, apesar dessa restrição, $P(G_m = g_{obs})$, sob a hipótese de independência, é capaz de diferenciar rotulações distintas satisfatórias daquelas que são indesejáveis. Tanto na simulação, quanto no problema de modelagem dos órgãos abdominais, rotulações ruins tiveram probabilidades ínfimas, enquanto rotulações mais próximas das ideais apresentaram probabilidades muito mais elevadas. Dessa forma, essa probabilidade se mostra adequada para ser otimizada e adotada como função-objetivo para reconhecer objetos estruturados segundo o modelo estatístico-relacional.

É importante ressaltar também que esse formalismo permite reconhecer diferentes elementos primitivos simultaneamente. Outros métodos existentes frequentemente exigem a aplicação consecutiva de um mesmo algoritmo que trata o caso binário, capaz apenas de reconhecer um único objeto e o fundo, até obter a segmentação do conjunto de objetos desejados.

5.1.3 *Ferramentas computacionais e implementação*

Para implementar a modelagem híbrida proposta, a aluna desenvolveu um arcabouço de programas na linguagem `C++`. Todos os métodos referentes a imagens foram construídos em torno da representação de imagens digitais nativa da biblioteca *Tivoli* (de *Traitement d'Images Volumiques*) desenvolvida no Depto. de Tratamento de Sinal e Imagens do *Institut Télécom ParisTech*. Também adotamos algumas funções dessa biblioteca, disponíveis para processamento de imagens volumétricas.

As estruturas de dados adotadas para implementar o conceito de grafo estatístico-relacional, bem como de grafos de observação, consistiram em extensões das estruturas de grafos nativas da biblioteca `boost`, para comportar os atributos e particularidades dos grafos propostos na tese. A implementação do formalismo probabilístico fez também uso de funções matemáticas e estatísticas da biblioteca *GNU Scientific Library*.

Em relação às imagens médicas, todas as segmentações e rotulações manuais dos órgãos abdominais nas imagens de treinamento foram realizadas com auxílio de um conjunto de ferramentas para visualização, manipulação, e tratamento de imagens médicas comumente utilizadas nos projetos conduzidos no *Institut Télécom ParisTech*, tais como *Mipav*, *3D Slicer*, *MedInria*, *Anatomist/Brainvisa*.

5.2 PERSPECTIVAS

A compreensão mais profunda dos problemas relacionados à modelagem híbrida para objetos estruturados, assim como do formalismo probabilístico, revelou novas questões a serem estudadas e aspectos a serem considerados futuramente, que relatamos nas seções a seguir.

5.2.1 *Desafios em relação ao modelo de objetos estruturados*

Como afirmamos na Seção 5.1.1, o processo de criação de um GER adequado para descrever um objeto estruturado compreende diferentes subproblemas, dentre os quais, a seleção de boas características de aparência e relacionais, e métodos para inferir parâmetros das distribuições de probabilidade que regem cada atributo.

Apesar desses subproblemas encontrarem soluções gratuitas no reconhecimento estatístico de padrões, muitos deles não são evidentes quando tratamos de relações. Por exemplo, a inferência e o aprendizado de boas relações para descrever um objeto estruturado complexo exige técnicas mais sofisticadas e não triviais, para as quais encontramos fundamentos na teoria de aprendizado estatístico-relacional [GT07]. Portanto, essa é uma área de interesse com potencial a ser explorado de acordo com o objeto estruturado a modelar.

Outra questão se refere às definições de relação espacial. Normalmente, essas relações são definidas segundo elementos da teoria de conjuntos difusos (*fuzzy set theory*), que são representadas por conjuntos difusos e suas respectivas funções de pertinência, que medem a pertinência de algum elemento à relação em questão. Assim, evocamos a questão de como trabalhar com estatísticas e probabilidades sobre essas medidas que não correspondem a um número, e como incorporá-las às estatísticas do modelo [Neu04, SB04]. Uma outra abordagem para essa questão é delegar a utilização das relações espaciais ao processo de reconhecimento, para que sejam adotadas como restrições às rotulações a serem exploradas a cada passo da otimização probabilística, e não diretamente no cálculo da probabilidade.

Finalmente, indicamos a extensão da modelagem ao uso de hipergrafos como representação de objetos estruturados, permitindo conectar diversos subconjuntos de nós de G_m, e não apenas pares de nós. Essa extensão permitiria explorar relações n-árias, com $n >= 3$, entre os elementos primitivos, como é o caso da relação "entre" [BCCJ06].

Por exemplo, no caso da aplicação em imagens RM, essa relação descreveria o fato de que "a coluna está entre o rim direito e o rim esquerdo". Ela também serviria para descrever a relação composta pelas asas direita e esquerda, e a fuselagem, no modelo de avião.

5.2.2 *Desafios em relação ao processo de reconhecimento*

Um dos principais trabalhos a serem efetuados na continuidade desta tese é concluir o estudo parcial sobre o formalismo de campos aleatórios Markovianos aplicado ao reconhecimento probabilístico de objetos estruturados. A possibilidade de explorar as informações de baixo nível oriundas da representação da imagem através dessa teoria e, simultaneamente, de considerar as informações de alto nível, correspondentes às relações espaciais introduzidas pelo modelo, demonstra autenticidade e um embasamento teórico significativo.

Ainda, podemos destacar os seguintes aspectos a serem investigados, referentes à etapa de otimização da probabilidade $P(G_m = g_{obs})$: estudo da possibilidade de considerar as probabilidades das características e as relações espaciais para restringir o espaço de possibilidades dos algoritmos de otimização; comparação de diferentes algoritmos de otimização em conjunto com a função-objetivo proposta.

5.2.3 *Desafios em relação à modelagem de órgãos computacionais*

A modelagem de órgãos abdominais não é uma tarefa simples, devido à complexidade das relações entre os órgãos, à elasticidade própria de alguns órgãos (como bexiga, estômago, intestinos), além da variabilidade das características interpessoais, ou mesmo das disparidades decorrentes de imagens de um mesmo paciente adquiridas em momentos distintos. Nesta tese, construímos um modelo computacional simplificado a partir de um subconjunto dos órgãos abdominais, mas que incorporou um conjunto de características de aparência e relacionais fidedignas aos conhecimentos anatômicos.

Num segundo momento, seria desejável a inclusão de novos órgãos no modelo, tais como bexiga, intestinos delgado e grosso, pâncreas, e vesícula biliar. Porem, é importante ressaltar que, para tanto, as imagens disponíveis devem ter uma resolução suficientemente grande para capturar as áreas correspondentes a órgãos menores.

Na modelagem realizada, consideramos imagens provenientes de um único tipo de equipamento de RM, de acordo com um único protocolo de aquisição. Seria interessante considerar também imagens de ressonância magnética heterogêneas quanto a esses aspectos no conjunto de treinamento, e na etapa de reconhecimento. Ainda, de modo mais amplo, seria interessante investigar meios de criar um modelo estatístico-relacional multimodalidade, para usufruir de possíveis in-

formações complementares oriundas de imagens de outra natureza, como as de tomografia computadorizada.

Evidentemente, a metodologia poderá ser aplicada também a outros problemas envolvendo objetos estruturados, para os quais informações estatísticas sejam relevantes, tais como os problemas de reconhecimento de cenas estruturadas de imagens naturais, ou segmentação de faces.

ANEXOS

A

LISTA DE PUBLICAÇÕES

Publicações referentes aos trabalhos colaborativos da aluna, que inspiraram o desenvolvimento da tese de doutorado desenvolvida:

- T. M. Paixão, A. B. V. Graciano, R. M. Cesar-Jr, e R. Hirata-Jr.: *A backmapping approach for graph-based object tracking*, Anais do Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens, SIBGRAPI 2008, 12-15 de outubro de 2008, páginas 45-52, ISSN 1530-1834, doi: http://dx.doi.org/10.1109/SIBGRAPI.2008.32.

- A. Noma, A. B. V. Graciano, L. A. Consularo, R. M. Cesar-Jr., e I. Bloch: *Inexact graph matching for segmentation and recognition of object parts*. Relatório Técnico, IME-USP, fevereiro de 2009.

- A. Noma, A. B. V. Graciano, L. A. Consularo, R. M. Cesar-Jr., e I. Bloch: *Interactive image segmentation by matching attributed relational graphs*, Pattern Recognition, volume 45, número 3, março de 2012, páginas 1159-1179, ISSN 0031-3203, doi: http://dx.doi.org/10.1016/j.patcog.2011.08.017.

- A. B. V. Graciano, T. M. Paixão, R. M. Cesar-Jr, e R. Hirata-Jr.: *Structural segmentation and labeling of object parts in digital video*, submetido à publicação *Signal, Image and Video Processing*, Springer-Verlag, março de 2012.

A Backmapping Approach for Graph-based Object Tracking

Thiago Meireles Paixão Ana Beatriz V. Graciano Roberto M. Cesar Jr.
Roberto Hirata Jr.
Institute of Mathematics and Statistics - University of São Paulo
Rua do Matão, 1010, 05508-090, São Paulo, Brazil
{thiagopx,abgv,cesar,hirata}@ime.usp.br

Abstract

Model-based methods play a central role to solve different problems in computer vision. A particular important class of such methods rely on graph models where an object is decomposed into a number of parts, each one being represented by a graph vertex. A graph model-based tracking algorithm has been recently introduced in which a model is generated for a given frame (reference frame) and used to track a target object in the subsequent ones. Because the view of an object changes along the video sequence, the solution updated the model using affine transformations. This paper proposes a different approach and improves the previous one in several ways. Firstly, instead of updating the model, each analyzed frame is backmapped to the model space, thus providing more robustness to the method because model parameters do not have to be modified. A different method for model generation based on user traces has also been implemented and used. This model generation approach is much simpler and user-friendly. Finally, a graph-matching algorithm that has been recently proposed is used for object tracking. This new algorithm is more efficient and leads to better matching results. Experimental results using synthetic and real sequences from the CAVIAR project are shown and discussed.

1 Introduction

Structural pattern recognition is based on the concept that each pattern is composed by a certain number of parts and the relations between them. For instance, a house may be decomposed as a roof on top of walls. A wall may have a door in it, which in its turn has a window beside it. The house parts are: roof, walls, door and window. The relations are *on top, in, beside*. There has been intense research on structural pattern recognition methods, particularly for computer vision applications [1, 4]. The present paper addresses the problem of tracking an object in a video sequence while recognizing its parts in each frame. A structural pattern recognition approach based on inexact graph matching is adopted.

There is a large literature on parts detection and recognition using graphs [3, 5]. The method discussed herein is based on a recent object tracking scheme [6] which, in turn, derives from the inexact matching method described in [2]. In this approach, the object of interest is decomposed into parts and represented as a graph. Each part of the model is associated to a graph vertex. Spatial relations are measured from the parts of the image and represented as graph edges linking the corresponding vertices. A similar graph is extracted from the image where the object parts should be segmented and recognized. These tasks are carried out by matching the vertices from the model to the input graph.

The present paper starts from the method described in [6] and improves it in a number of different ways. The tracking and recognition results are better and obtained in a much more efficient way. Model generation is carried out from imprecise traces made by a user through a specially-designed GUI. The model update strategy has been replaced by a model-space oriented concept: the frame to be recognized is backmapped to the model space, which leads to more robust results since the model parameters do not have to be re-estimated for each frame. A new matching algorithm, recently introduced in [7] for static image segmentation, has been adapted and employed, leading to better and faster results. Finally, different pre-processing steps based on motion estimation and connected component analysis have been created. These aspects constitute the main original contributions of the present paper.

This paper is organized as follows. Section 2 reviews the main concepts of the approach, thus providing an overview of the method. The main contributions of the current paper are presented in Section 3, which is followed by the experimental results in Section 4. The paper is concluded in Section 5.

2 Graph-based methodology overview

This section reviews the mathematical modeling used in [6] for object tracking. The basic idea behind the method is to identify the target object in a given frame (reference frame) and then to track it by recognizing and mapping its parts in the subsequent frames. Therefore, object identification in the reference frame is equivalent to identifying its parts. This task may be carried out interactively (user-oriented) or by means of automatic detection procedures (e.g. face detection methods for face tracking and recognition applications). This step is known as *model generation* and is addressed in Section 3.1. Both object model and subsequent frames are similarly represented by attributed relational graphs (ARG) [9]. An ARG is actually a graph in which attribute vectors are assigned to vertices and to edges. Such vectors are responsible for adding relevant problem information to a graph data structure, since they hold symbolic properties and features related to the nodes and edges they are assigned to. We will henceforth refer to the *model graph* simply as *model*. On the other hand, the *input graph* is associated to the target frame where the object of interest should be tracked / recognized. Figure 1 provides an overview of the method. A model G_m is generated in the reference frame (F_0). Subsequent frames $F_1, F_2, \ldots, F_n$ are analyzed so that input graphs $G_i^1, G_i^2, \ldots, G_i^n$ are also generated. Model and input graphs are matched, thus resulting in the tracked objects with recognized parts (indicated as R_i in the figure).

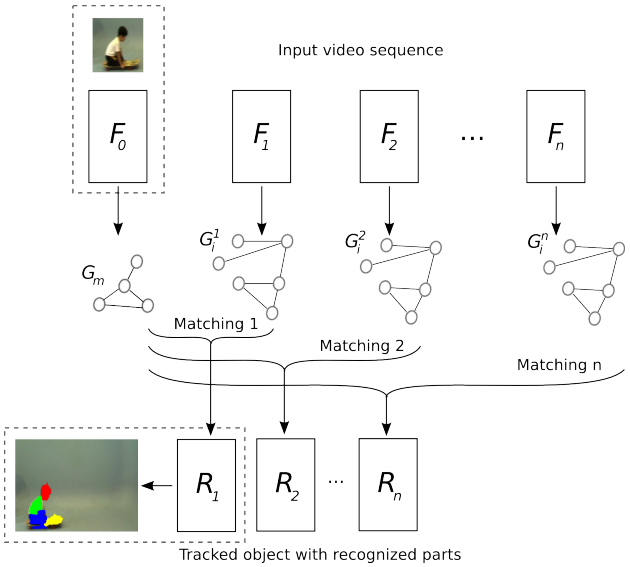


Figure 1. Overview of tracking process.

More formally, an ARG is a graph with feature vectors defined by a tuple $G = (V, E, \mu, \nu)$, where V denotes the vertices and E the edges. V is associated to object parts, whereas edges represent spatial relations between them. $\forall v \in V$, there is an associated *object attribute vector* $\mu : V \rightarrow \mathbb{R}^p$. Analogously, $\forall e \in E$, there is an associated *relational attribute vector* $\nu : E \rightarrow \mathbb{R}^q$. The values p and q indicate the number of vertex and edge attributes, respectively. Adjacency graphs [8] have been used to define the graph topology.

Let $G = (V, E, \mu, \nu)$ be an ARG, $v, w \in V$ be two vertices and $a = (v, w) \in E$ and $a' = (w, v) \in E$ be two edges. The *object* attribute vector $\mu(v)$ is defined as:

$$\mu(v) = (RGB(v), c(v)). \tag{1}$$

The term $RGB(v)$ is a tuple $(r(v), g(v), b(v))$ of the normalized average RGB-levels of the image region associated to vertex v, whereas $c(v)$ indicates the centroid coordinates of the region. The *relational* attribute vector is defined as:

$$\nu(v, w) = (\overrightarrow{v}), \text{ with } \overrightarrow{v} = \frac{\overrightarrow{(c(w) - c(v))}}{d_{max}}. \tag{2}$$

where d_{max} is the maximum distance between two points in the image.

The input ARG G_i is generated from the watershed partition [10] of an input frame. Each watershed region is associated to an input ARG vertex and its object attributes vector is calculated from the corresponding watershed region (i.e. average RGB tuple and centroid). The watershed partition also helps to define the ARG topology, which is created as an adjacency graph based on the watershed basins neighborhood. It is also necessary to calculate the model G_m, which is explained in Section 3.1.

A solution to the tracking/recognition problem is a mapping from V_i to V_m [6]. Because $|V_i|$ is usually much larger than $|V_m|$, a suitable homomorphism between G_i and G_m should map distinct vertices of G_i onto a single vertex of G_m, which corresponds to merging coherent input regions. An *association graph* $\tilde{G}_A$ between G_i and G_m is defined as the complete graph $\tilde{G}_A = (V_A, E_A)$, where $V_A = V_i \times V_m$ and $E_A = E_i \times E_m$. Thus, $\tilde{G}_A$ is a graph representation of all possible mappings from G_i to G_m. Particularly, homomorphisms between G_i and G_m are a family of such possible mappings. A graph homomorphism h between G_i and G_m is a mapping $h\colon V_i \rightarrow V_m$ such that $\forall a_1 \in V_i, \forall b_1 \in V_i, (a_1, b_1) \in E \Rightarrow (h(a_1), h(b_1)) \in E_m$. This definition assumes that all vertices in G_i should be mapped to G_m.

As proposed in [2], a solution for finding a homomorphism between G_i and G_m may be defined as a complete subgraph $\tilde{G}_S = (V_S, E_S)$ from the association graph $\tilde{G}_A$, in which $V_S = \{(a_1, a_2), a_1 \in V_i, a_2 \in V_m\}$ such that $\forall a_1 \in V_i, \exists a_2 \in V_m, (a_1, a_2) \in V_S$, and $\forall (a_1, a_2) \in$

$V_S, \forall(a_1', a_2') \in V_S, a_1 = a_1' \Rightarrow a_2 = a_2'$, assuring that each vertex from the input ARG corresponds to exactly one vertex of the model ARG and $|V_S| = |V_i|$. Such a solution only considers the structures of G_i and G_m, giving rise to many possible homomorphisms between both graphs. In order to find a suitable solution, a cost function is defined, so that the search may be expressed as an optimization problem.

Consider again G_i and G_m, as well as vertices $a_1, b_1 \in V_i$, $a_2, b_2 \in V_m$ and edges $e_1 \in E_i$, and $e_2 \in E_m$. Let $\tilde{G}_S$ be a suitable subgraph of the association graph $\tilde{G}_A$ that represents a valid solution. The adopted cost function is defined as:

$$f(\tilde{G}_S) = \frac{\alpha}{|V_S|} \sum_{(a_1,a_2)\in V_S} c_V(a_1, a_2) + \frac{(1-\alpha)}{|E_S|} \sum_{e \in E_S} c_E(e). \tag{3}$$

f is a weighted sum of object (c_V) and structural (c_E) properties. The cost functions c_V and c_E, defined in [7] and adopted in this paper, consist of dissimilarity measures between vertices and edges, respectively. Thus, compatible pairs of vertices or edges present small dissimilarity value and contribute to minimizing f.

3 Proposed approach

This section presents the improved methodology to object tracking using graph matching. We first present the new tool for model generation, and the strategy to facilitate the segmentation algorithm. We propose a way to approximate the object for detection and then a new paradigm that maps the input graph back to the model space.

3.1 Model generation

The model is an ARG that represents object parts and background of an object neighborhood. In Section 2, we described the ARG generation process. Model generation is a particular case of ARG generation. In [7], the model generation process is presented to segment static images. We adapted this procedure by introducing background labeling, which is fundamental for object tracking. Basically, a model is generated by an interactive process in which the user makes some traces associating object parts to specific labels. If a labeled trace intersects a watershed region, a correspondent new labeled vertex is added to the ARG. Figure 2 depicts this step.

A specific tool including a GUI (Figure 3) to control this process has been developed using the platform C++/GTKmm [1]. This tool is integrated with Intel OpenCV library [2], which implements the tracking methodology.

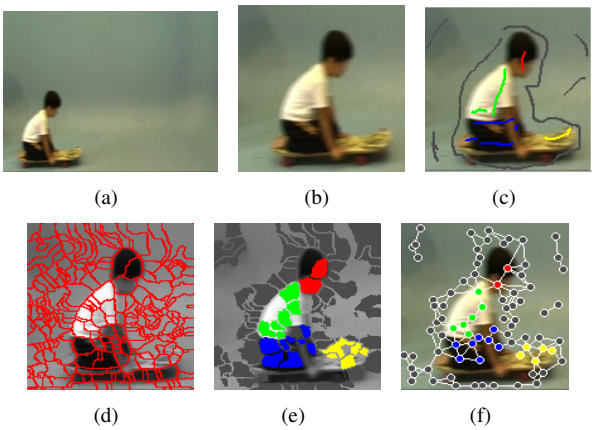


Figure 2. Model generation steps: (a) reference frame; (b) region of interest; (c) user-defined traces; (d) watershed partition; (e) intersection of watershed regions with user-defined traces; (f) model superposed to original image.

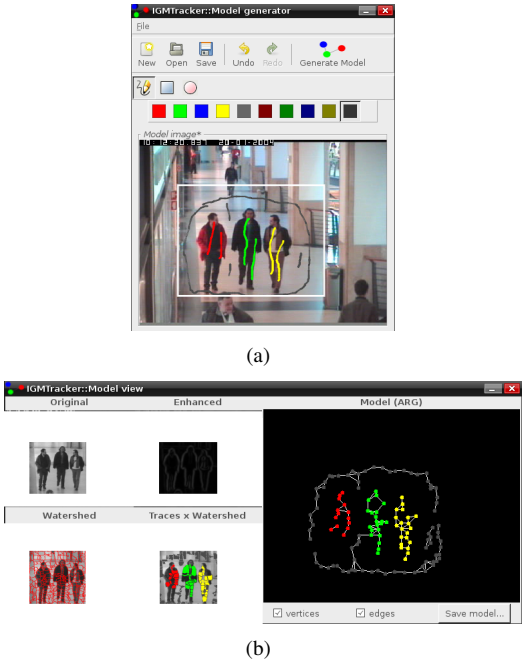


Figure 3. Model generator: (a) ROI and traces defined by user; (b) resulting model.

[1] http://www.gtkmm.org

[2] http://sourceforge.net/projects/opencvlibrary

3.2 Pre-processing

Yilmaz et al. [11] list four ways to detect objects: point detectors, background subtraction, segmentation and supervised learning methods. We based our method in a restricted area of segmentation, i.e. segmentation is performed inside a region of interest (ROI) for the sake of computational savings and object segmentation quality. The pre-processing step aims at obtaining a dilated object silhouette as a ROI and control points to calculate object motion.

Initially, the user defines a rectangular ROI, which may not represent the object shape properly. Therefore, we compute an input ARG from the original ROI and segment this region by matching against the model. The result includes classified object parts as well as the classified background region. The object silhouette is the union of object parts whereas the strategic control points are the centroids of parts. This process is illustrated in Figure 4.

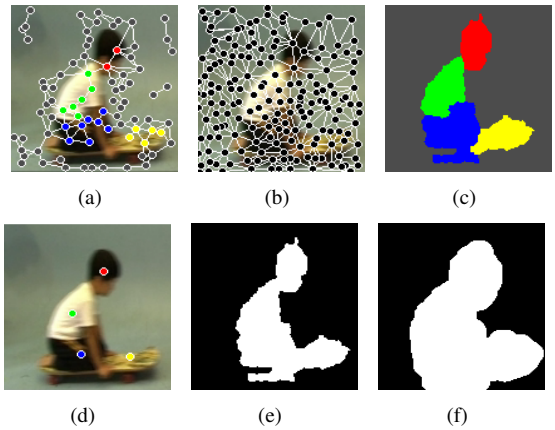


Figure 4. Pre-processing steps: (a) model superposed to rectangular ROI; (b) input graph computed inside a rectangular ROI; (c) segmentation; (d) centroids of object parts for motion estimation purposes; (e) object silhouette; (f) dilated object silhouette.

3.3 Approximate object detection

Before the final object detection through model-based segmentation, we have proposed an *approximate object detection* pre-stage to simplify the input graph extraction and the graph-matching computing. We estimate an affine transform T_i that maps an object of an input frame i onto the object in reference frame. T_i^{-1} denotes the inverse of T_i. Let S be a binary image of the dilated object silhouette obtained in the pre-processing step. Therefore, $T_i^{-1}(S)$ denotes the region of interest of the input frame i. In fact, the transformation T_i can only be estimated after segmentation of frame i because the segmentation provides control points that match those in pre-processing step. These control points allow us to calculate a proper affine transformation (see Section 3.4).

We used a dilated version of the silhouette because even if T_i is not accurate, $T_i^{-1}(S)$ still will include the whole object. At this point, we emphasize that the object is not fully detected, because the region may include non-object parts that need to be classified. $T_i(S)$ plays the role of a mask for input graph generation in order to save computational processing. It helps to eliminate other moving objects that may lie on any part of the scene.

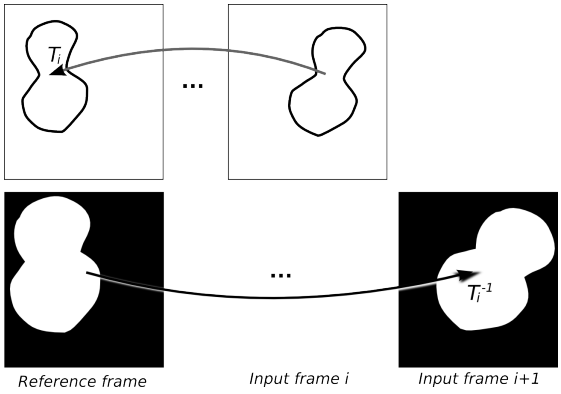


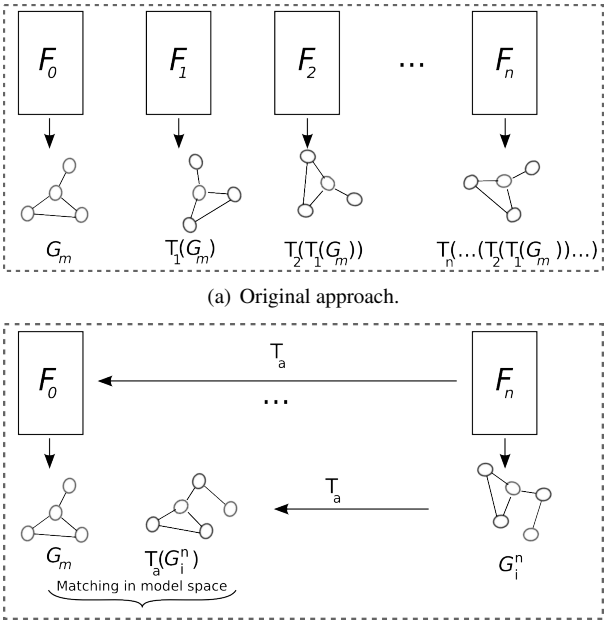
Figure 5. Approximate object detection. In the first row, the transformation T maps the object of input frame i onto the object of the reference frame. In the second row, we use T^{-1} to map the dilated object silhouette onto the approximate dilated silhouette of input frame $i+1$.

3.4 Backmapping

The key idea behind our tracking methodology is to backmap an object onto the reference frame for segmentation. In this work, backmapping is defined as the task of estimating an affine transformation that maps an input graph back to the model space. This procedure and the segmentation step are strongly interdependent. During the second input frame processing, the respective input graph is not classified and is slightly misaligned with respect to the model. Thus, the segmentation step precedes the backmapping. After segmentation, the backmapping is estimated and applied to the input graph of the next frame. A well-performed backmapping depends on a well-performed segmentation and vice versa.

This approach differs from the original method which is based on *model update*, i.e. the model is mapped onto the

object of the current frame, whereas our approach holds the model fixed in order to prevent accumulation of errors in backmapping estimation and therefore model degeneration. Figure 6 confronts other two approaches.

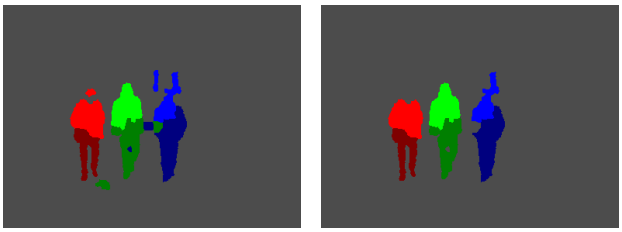


(a) Original approach.

(b) New approach.

Figure 6. Original x new approach. In the original approach the model suffers sequential transformations that may degenerate the model. In the new approach the model is fixed and input graph is backmapped.

We assume that an object part is a connected component. After the segmentation step, we filter the resulting image of selecting the largest connected component for each part (Figure 7).



(a) (b)

Figure 7. Connected components heuristic: (a) raw segmentation; (b) segmentation after connected component filtering.

The last step is to estimate the backmapping affine transformation. We take the centroids of each classified object part and we build a linear system to recover the transformation matrix. The affine transformation is given by six parameters, so the segmented input frame, and also the reference frame, must present at least three object parts.

4 Experimental results

The algorithms have been implemented using C++ and Intel OpenCV library. The timings reported here refer to a Celeron machine (1.73GHz) with 1GB of RAM. The dataset has three MPEG sequences (320×240 pixels/frame) and frame rate of 30fps. Segmentation was performed with $\alpha = 0.4$ (weight for vertex cost, see Section 2) and $\delta = 0.6$ (weight for modulus edge cost [7]). Results present tracked sets with recognized parts and approximate object detection. The time consumption for the three cases is about 10 seconds for model generation, 13 seconds for the preprocessing step and 20 seconds per processed frame.

In the first sequence, a boy performs a translation from left to right of scene. The boy's body, skate board and some background have been labeled according to the Figure 8(a). The result (Figure 9) shows 5 sequence frames sampled in 5-spaced intervals, from a total of 36 frames. We achieved a good visual performance in this sequence, with just very small variations along the frames due to excessive blur and low resolution image.

The second sequence is from a rotation of a doll. The doll's body and a background region have been marked according to Figure 8(b). In this case, the watershed gives rise to only one background region. This is a problem because the whole background is represented by just a single vertex. If that vertex is misclassified, all the background is misclassified. Thus, new vertices are generated around object by splitting of background into a $p \times q$-square grid ($p = q = 25$ pixels in this sequence). The result (Figure 10) shows 5 sequence frames sampled in five 15-spaced intervals, from a total of 60 frames (rotation of 60 degrees). In column 4 of Figure 10(b), the doll's right arm (reference of the reader) was misclassified as background due to a post-processing filter. If on the one hand this post-processing helps to maintain a good visual performance, on the other, objects may disappear if they disconnect from the main part. In the next frame, the tracking algorithm recovers from the mistake.

The third sequence is from the CAVIAR project [3]. There are three people walking from top to bottom of the image causing a zoom effect. We modeled the three people as a unique object (Figure 8(c) depicts object and background modeling). This sequence includes challenging elements such as other moving people, shadows, heterogeneous background and poor object resolution in reference frame. Fig-

[3] EC Funded CAVIAR project/IST 2001 37540, found at URL: http://homepages.inf.ed.ac.uk/rbf/CAVIAR/

ure 11 shows some results. Our approximate object detection strategy eliminates non-target objects. In some frames, dark strips of the floor was merged to dark pants of the three targets, a limitation due to the way that the watershed algorithm is being used. The advantage of structural approach is that artifacts, like other people walking on scene, do not introduce errors in the tracking if they are not very close to the tracked object. On the fourth column of the second row, approximate object detection estimates a rotation that does not occur in video sequence. The cause is the loss of few object parts, therefore a bad estimation of motion control points (object parts centroids). Common effects of bad estimation is rotation and, in some cases (depending on object symmetry), scaling.

(a) (b) (c)

Figure 8. Regions of interest and traces for three video sequences.

5 Concluding remarks

Recent advances on video analysis show that Structural Pattern Recognition may be a promising way to approach the object tracking problem. A previous work models the objects to be tracked as a graph with attributes and, for each frame, the model is updated using affine transforms. In this paper, we proposed a new paradigm where the model is not updated for the subsequent frames but it is the object that is backmapped to the original model, simplifying the algorithm. The graph-matching algorithm involved in this process has also been changed to the one presented in [7]. In relation to the previous one, this one has decreased the runtime complexity from $O(|V_g|^{|V_i|})$ to $O(|V_g| * |V_i|)$, drastically dropping the amount of time taken to process a given video. In practice, when tested[4] against the goose sequence [6], it has taken 5 minutes (in comparison to 78 minutes from the previous work), with visually similar results. Finally, the initial model generation is done now with a user-friendly interface. The approach has been implemented, tested (to certify against implementation mistakes) and experimented in some video sequences, three of them were reported in this work (one of them is a surveillance video used to test motion segmentation algorithms). The results are very promising and show significant improvements in relation to the previous approach.

[4]experiment available on www.vision.ime.usp.br/ thiagopx/sibgrapi/

Acknowledgements.

The authors are grateful to FAPESP, CNPq and CAPES for partial (Roberto M. Cesar Jr. and Roberto Hirata Jr.) and full financial support (Thiago M. Paixão and Ana B. V. Graciano).

References

[1] H. Bunke. Recent developments in graph matching. In *ICPR*, pages 2117–2124, 2000.

[2] R. Cesar-Jr, E. Bengoetxea, P. Larranaga, and I. Bloch. Inexact graph matching for model-based recognition: Evaluation and comparison of optimization algorithms. *Pattern Recognition*, 38(11):2099–2113, November 2005.

[3] Y. Chen, L. Zhu, C. Lin, A. Yuille, and H. Zhang. Rapid inference on a novel and/or graph for object detection, segmentation and parsing. In J. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 289–296. MIT Press, Cambridge, MA, 2008.

[4] D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty years of graph matching in pattern recognition. *IJPRAI*, 18(3):265–298, 2004.

[5] P. F. Felzenszwalb and J. D. Schwartz. Hierarchical matching of deformable shapes. In *CVPR*. IEEE Computer Society, 2007.

[6] A. B. V. Graciano, R. M. Cesar-Jr., and I. Bloch. Graph-based object tracking using structural pattern recognition. In *SIBGRAPI '07: Proceedings of the XX Brazilian Symposium on Computer Graphics and Image Processing*, pages 179–186, Washington, DC, USA, 2007. IEEE Computer Society.

[7] A. Noma, A. B. V. Graciano, L. A. Consularo, R. M. Cesar-Jr, and I. Bloch. A new algorithm for interactive structural image segmentation. 2008.

[8] T. Pavlidis. Algorithms for shape analysis of contours and waveforms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-2(4):301–312, 1980.

[9] R. J. Schalkoff. *Pattern recognition: statistical, structural and neural approaches*. John Wiley & Sons, Inc., New York, NY, USA, 1991.

[10] L. Vincent and P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, 13(6):583–598, 1991.

[11] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38(4):13, 2006.

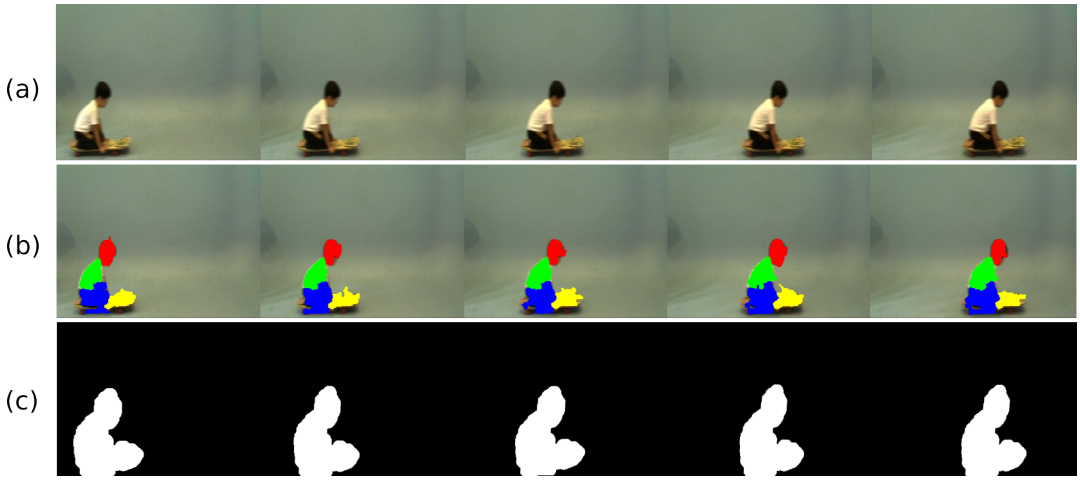


Figure 9. Results of the first experiment.

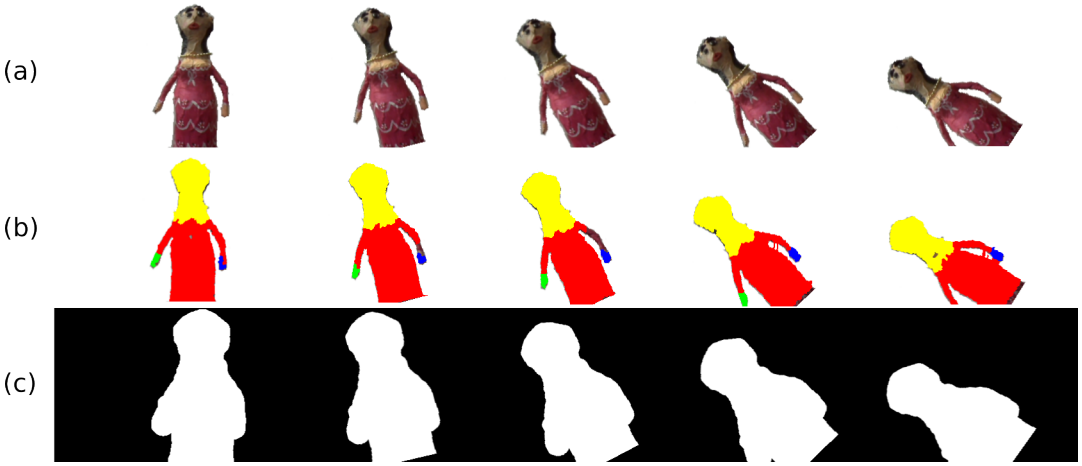


Figure 10. Results of the second experiment.

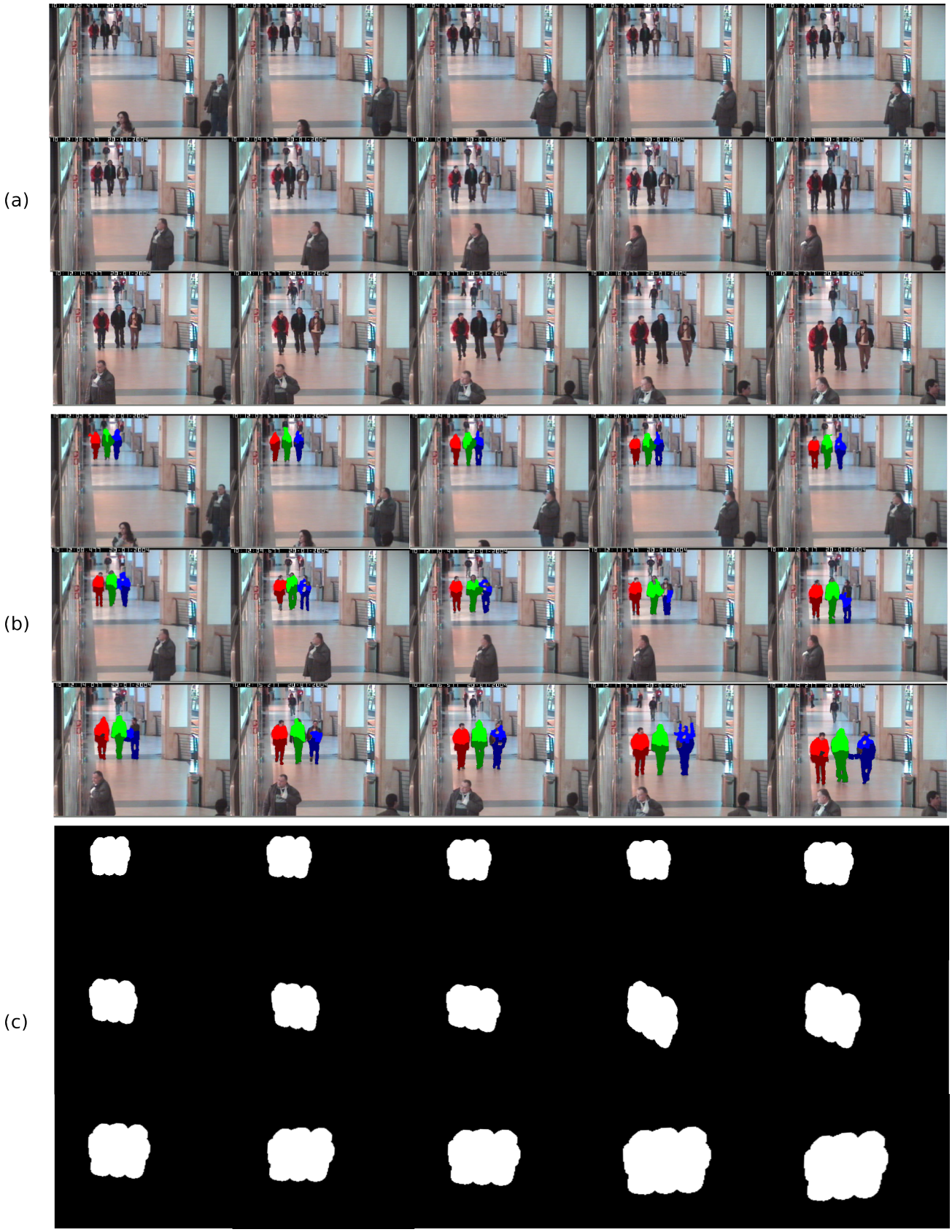


Figure 11. Results of the third experiment.

Inexact Graph Matching for Segmentation and Recognition of Object Parts

Alexandre Noma, Ana B. V. Graciano, Roberto M. Cesar Jr.,
Luis A. Consularo [*], Isabelle Bloch [†]

Technical Report
Dept. of Computer Science, Institute of Mathematics and Statistics,
University of São Paulo, São Paulo, Brazil
e-mails: {noma,cesar,abvg}@ime.usp.br

February 13, 2009

Abstract

This article presents a novel graph matching technique for segmenting and recognizing object parts in images according to a model. Both model and input images are represented by means of attributed relational graphs which convey appearance features through object attributes and structural properties through relational attributes. Since the model and input graphs might present topological differences, a new structure called the *deformation* graph is introduced. The segmentation and recognition process is achieved through a new graph matching algorithm based on the deformation graph, which corresponds to labelling the input graph in order to minimize the deformations introduced in the model when it is updated with input information. This method is applied to the problem of multiple label segmentation and has shown to be fast with successful experimental results on both natural color images and video. Experiments have also allowed the evaluation of model robustness and reuse when segmenting different images using the same generated graph model to recognize similar objects in each of them, thus minimizing user intervention. The proposed technique has been implemented as an open-source application available at http://structuralsegm.sourceforge.net/.

1 Introduction

Segmentation and recognition of objects in digital images are essential steps in many important imaging applications. Solutions to such problems cover a wide range of techniques, among which, those based on graphs. A graph is a

[*] L.A. Consularo was with UNIMEP and is now with the TSE - Tribunal Superior Eleitoral, Brasília, Brazil, Brazil (e-mail: consularo@gmail.com).

[†] I. Bloch is with the Signal and Image Processing Department, TELECOM ParisTech (ENST), CNRS UMR 5141 LTCI, Paris, France (e-mail: isabelle.bloch@enst.fr).

powerful data structure for object representation, since its vertices and edges can be related, respectively, to pictorial elements and their relations and structure. Graphs may easily describe objects in terms of their component parts, being especially suitable when dealing with non-rigid objects such as facial features or the human body [1–3], or any kind of structured object in general [4]. When attributes are assigned to each vertex and to each edge of a graph, representing appearance and relational features, the graph is usually called an *attributed relational graph* (ARG).

The literature on graph-based image segmentation is extensive and introduces important methods, such as the Image Foresting Transform (IFT) [5], graph-cuts and Markov-Random-Fields-based (MRF) methods [6–9], amongst others. According to the resulting image partition, segmentation methods might be classified as binary, when only a foreground-background image is obtained, or as multi-label, when more than two meaningful image regions are produced. Graph-cut methods [6, 7, 9] are an important example of the binary segmentation case. Although not linear in the number of pixels of the input image, the graph-cut based approaches are reported to have very reasonable running times [10]. However, an extension of such methods to the multi-label problem is reduced to the multiway-cut problem [11], which is NP-Hard, requiring heuristics to obtain a solution and additional computational effort [12]. Grady [13] has also proposed a multi-label interactive image segmentation algorithm, in which the input image is viewed as a graph and the task is to determine the probability that a random walker starting at each unlabeled pixel will first reach one of the pre-labeled pixels.

Another class of graph-based segmentation methods has also been proposed in the literature [1, 14–18], usually under an object recognition perspective. In most cases, some supervised information about the target object(s) is expressed by means of a prototype model of the objects of interest, and segmentation is thus seen as a graph-matching problem between two graphs representing the input and the model. The matching step implies a mapping from the input vertices to their corresponding model vertices and, according to the restrictions imposed over such mappings, the algorithms may be classified either as exact or inexact (also known as error-tolerant).

The first category usually searches for strict correspondences between the sets of vertices and edges, accepting isomorphisms [19], sub-graph isomorphisms [20], or at most homomorphisms. All these guarantee that at least all edge relations shall be preserved. On the other hand, the inexact category [19, 21–25] allows approximate solutions when an exact mapping is not available, which is usually done through an optimization procedure which assigns a cost value to each solution and heuristics which search for minimizing or maximizing such values. These procedures might evaluate various aspects of the graphs, such as operations of vertex or edge inclusion or removal, vertex or edge (dis)similarities, among others.

Although the model and input images are related, their ARG representations often present different topologies due to the distinct natures of input and model partitioning, which produces different number of vertices and edges (Fig. 1). Such differences imply difficulties to determine a suitable mapping between the input and model graphs, as well as high computational

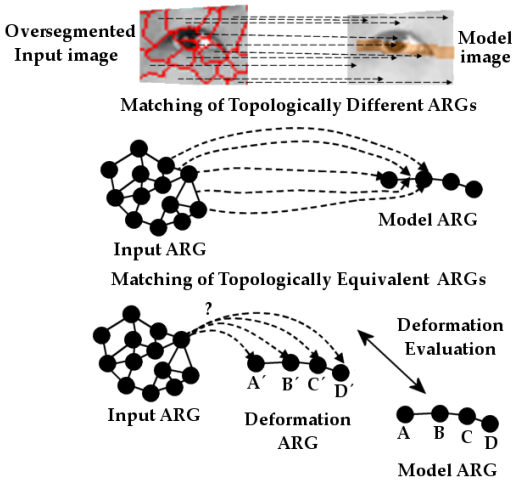


Figure 1: Matching of two topologically different attributed relational graphs versus two topologically equivalent ones under a deformation point of view.

cost. There exists a combinatorial number of possible graph matchings and the optimization procedure not only has to consider vertex similarities, but it also has to evaluate various structural matching configurations in order to avoid fragmented regions and preserve the borders of the objects. The structural information is used to rule out those "configurations" which are not plausible for a final segmentation. However, this might be misleading when both topologies are distinct and cause the method to look down on potential solutions, as shown in Fig. 1. For instance, consider an input edge connecting two input vertices which represent adjacent oversegmented regions related to two distinct objects (e.g. a subregion of the iris and an adjacent subregion of the sclera in Fig. 1). Consider also a model edge connecting model vertices which represent the same previous objects (e.g. the iris and the sclera). Although such a match is suitable in theory, the edges are quite different and may thus be evaluated as a bad solution.

In order to take structure into account and yet avoid the problem of directly matching two topologically different graphs, this paper introduces a novel inexact graph matching algorithm which deals with this matter by viewing each possible matching from an input ARG vertex to a model ARG vertex as a local deformation of the model graph. Such a deformation is expressed by means of the concept of the *deformation* ARG, which represents a distorted version of the model ARG that preserves its topological properties (i.e., its structure) while merging the attributes of a model vertex with the attributes of an input vertex.

These ideas are a natural evolution of the methods proposed in [3, 26]. A first attempt to segment static images through inexact graph matching was taken in [3]. The method aimed at segmenting facial features according to a manually-created face model image. Both model and input were represented by complete attributed relational graphs, i.e., ARGs with a complete topology, which were then matched in the recognition step. Although these

graphs were complete, they were often considerably distinct since the input one was derived from an oversegmentation and the model one, from a simplified partition of the target regions. However, these differences were not explicitly taken into account. Three inexact approaches considering structure have been evaluated: tree-search, genetic and estimation of distribution algorithms. These optimization techniques have been evaluated in terms of computational efficiency and result accuracy, revealing the best performance for the tree-search approach. Although this work presented successful results, the algorithms were time-consuming and the usage of complete graphs did not necessarily convey a compact representation of object structure.

Thus, the method in [26] proposed a new graph matching algorithm applied to the interactive segmentation problem. The technique aggregated user interaction in order to create a model of the objects to be segmented based upon the user scribbles, similarly to what is done in the present work. Then, segmentation was performed through the proposed graph matching algorithm, which was based on the Sequential Forward Search (SFS) strategy for feature selection [27], and entailed structure in its cost function. The algorithm dealt with differences in graph topology when searching for a mapping: edges from the input were compared to those from the model according to a dissimilarity measure that considers four special topology-match cases in order to avoid the aforementioned structural problems. Each case consisted of one possible combination of presence or absence of an edge with respect to the model graph. However, this approach to the topological discrepancy problem did not cover all possible cases and segmentation could be impaired under certain circumstances.

Regarding the segmentation and recognition issues, the present paper proposes a new algorithm for multi-label object segmentation according to an interactively created part-based model using ARGs as the underlying representation. An object (Fig. 2 (a)) is considered to be a set of parts (subsets of pixels of an image) and their relations. An object model image presenting the relevant object(s) and composing parts of interest is defined by the user according to scribbles drawn over this image (Fig. 2 (b)), as described in [26]. However, the model image could as well be obtained manually as in [3], or automatically inferred as in [28]. Then, the input image is oversegmented, giving rise to two graphs: a model graph, obtained from the oversegmentation of the input image, which includes only the regions that intercept the user traces; and an input graph, which includes all the regions from the oversegmentation. Thus, user-defined traces are used to select a subset of regions of the oversegmented image. Finally, the recognition step is solved through a novel graph matching procedure proposed herein. The resulting image is a partition of the input, labelled according to the model (Fig. 2 (c)). Segmentation and recognition output have been assessed through tests performed over sample images from public libraries and sample frames from two video sequences. Results illustrating the reusability of a single model applied to similar objects in order to justify the model robustness are also presented. All these elements are the main original contribution of the present paper.

This paper is organized as follows. Section 2 presents the formal defini-

(a) (b) (c)

Figure 2: Sample images and the definition of objects and parts in this paper: (a) an object might be a whole scene subdivided into parts (buffalos, background), or a single entity (a person) subdivided into meaningful parts (face, clothing, hair); (b) all part-models are defined by the user through traces drawn over the regions of interest, and each trace color identifies a single object/part; (c) these colors also identify the recognized objects of interest in the segmented images.

tion of attributed relational graphs for image representation adopted in this paper, as well as an overview of the segmentation method proposed herein. Section 3 explains the proposed graph matching algorithm for the purpose of segmentation and recognition, which is based on an optimization technique, described in Section 4. Experimental results are described in Section 5, in which the benefits of the proposed method are highlighted. Finally, a few conclusions, as well as suggestions for future work, are given in Section 6.

2 Methodology overview

2.1 Graph-based representation

In this paper, an *attributed relational graph* (ARG) is a directed graph formally expressed as a tuple $G = (V, E, \mu, \nu)$, where V stands for the set of vertices of G and $E \subseteq V \times V$, its set of edges. Typically, a vertex represents a single image region (subset of image pixels) and an edge is created between vertices representing two image regions. $\mu : V \rightarrow L_V$ assigns an *object* attribute vector to each vertex of V, whereas $\nu : E \rightarrow L_E$ assigns a *relational* attribute vector to each edge of E. $|V|$ denotes the number of vertices in V, while $|E|$ denotes the number of edges in E. Similarly, $|L_V|$ and $|L_E|$ correspond to the number of attributes which compose the vectors μ and ν, respectively. $E(v)$ denotes the set of all directed edges with an endpoint at $v \in V$, i.e. $E(v) = \{e \in E : e = (v, w) \text{ or } e = (w, v), w \in V\}$.

For the purpose of the segmentation method, three instances of such ARGs are considered: an *input* ARG $G_i = (V_i, E_i, \mu_i, \nu_i)$, derived from the input image, a *model* ARG $G_m = (V_m, E_m, \mu_m, \nu_m)$, representing the objects of interest selected by the user, and a *deformation* ARG $G_d = (V_d, E_d, \mu_d, \nu_d)$, used as an auxiliary data structure for measuring deformations implied in the model when matching a vertex $v \in V_i$ to another $w \in V_m$. Subscripts shall be used to denote the corresponding graph, e.g.

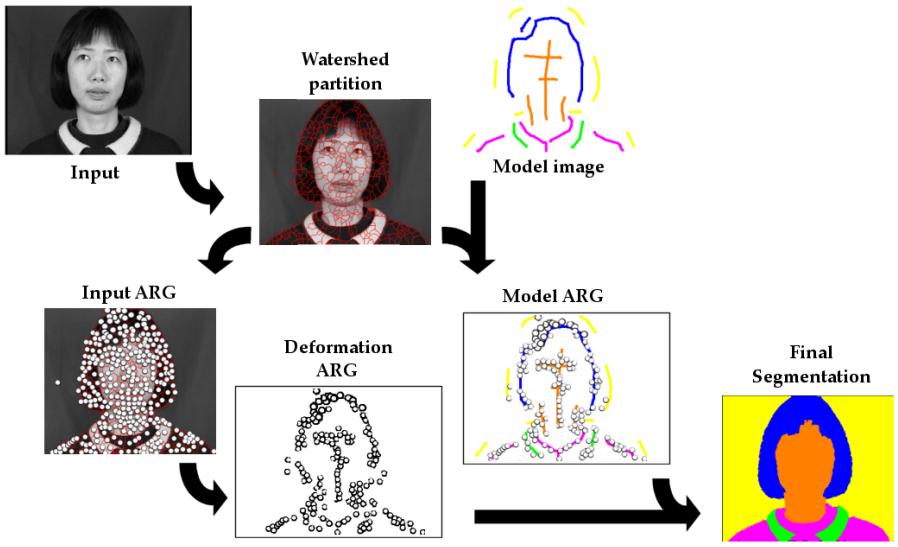


Figure 3: Overview of the methodology steps. The ARGs are depicted only with their vertices for better visualization.

$v_i \in V_i$ denotes a vertex of G_i, whereas $e_i = (v_i, w_i) \in E_i$ denotes an edge of G_i. Similar notations are used for G_m and G_d. Note that, when referring to a graph G_d, an association between a given pair of vertices v_i and v_m is always implicit.

- $\mu(v)[j]$ $(j = 1, \ldots, |L_V|)$ and $\nu(e)[k]$ $(k = 1, \ldots, |L_E|)$ refer to the "j-th" and "k-th" attributes of the respective vectors $\mu(v)$ and $\nu(e)$ associated to a vertex v and an edge e, respectively;
- $\mu'_d(v_d)[j]$ and $\nu'_d(e_d)[k]$ refer to the updated attribute vectors of a vertex $v_d \in G_d$;
- σ^j_μ and σ^k_ν are *update* functions used to compute attributes of a vertex belonging to G_d;
- δ^j_V and δ^k_E are *dissimilarity* functions used to compare the respective object and relational attributes of two given graphs.

2.2 Proposed framework

The segmentation process is depicted step-by-step in Fig. 3. First, a model image must be created and it should indicate a labelling of all the objects to be segmented and their parts of interest. In this paper, the model image is created interactively as described in [26]: given an input image to be segmented, the user first points out the target objects by placing traces over the input, thus creating a model image in which each color identifies an object (or object part) of interest. Next, an oversegmentation is performed using the watershed algorithm [29] in order to obtain a partition image in which the expected contours of each object are present.

This oversegmented image is used to create both an input ARG G_i and a model ARG G_m. The first is obtained in the following way: each watershed

region gives rise to a vertex and its attributes, whereas adjacent regions lead to an edge and its respective attributes. G_m is obtained similarly, but only those watershed regions which intercept the user-defined traces result in a model vertex. The model edges, however, are created according to the desired graph topology (in the present work, complete and triangulated graphs are used). Since it is desirable that the user inputs little effort when segmenting an image, only a few scribbles are expected over the image, thus implying a considerable difference in the number of vertices present in the input and model graphs (Fig. 3). The final segmentation should be a mapping of all $v_i \in V_i$ to vertices in V_m such that input vertices related to image regions corresponding to the same model object should be assigned to the same model vertex. This is equivalent to merging regions of an oversegmented object into a single region. The mapping of vertices from G_i to those in G_m characterizes an inexact graph matching problem [14, 15]. Although many mappings exist, a desirable solution should correspond to an image partition as similar to that defined by the model image as possible.

In order to guarantee that the final mapping follows the model ARG topology, the *deformation* ARG G_d is introduced. This graph is initialized as a copy of G_m and used to evaluate the local deformation effect produced by a given association between a vertex $v_i \in V_i$ and a vertex $v_m \in V_m$ (Fig. 4). This effect is computed through the update of the object attributes of vertex v_d, a clone of v_m, with the object attributes of v_i, as if the regions represented by such vertices were the same. Also, the relational attributes of all edges directly connected to v_d are updated. Note that this update preserves the topology of G_d, solely affecting the values of the object and relational attributes. Finally, this deformed version of the model graph is compared to the original one through a cost function. A desirable solution should minimize this cost and the final classification of a given v_i will be the vertex v_m that corresponds to the minimum cost association. In the next section, we discuss how these deformations are computed and used by the graph matching procedure.

3 The graph matching algorithm for model-based image segmentation

A segmentation of the input image according to the model under the graph-based representation is a solution for the graph matching problem between G_i and G_m, characterized as a mapping $f : V_i \rightarrow V_m$. This implies assigning a corresponding model vertex to each input vertex. Clearly, there are $|V_m|$ possible assignments for each input vertex and the decision of which to choose depends on an optimization procedure.

Let G_d be an ARG initially equal to G_m, i.e., $V_d = V_m$ and $E_d = E_m$. Let also v_d and v_m be two corresponding vertices in G_d and G_m respectively, with $\mu_d(v_d) = \mu_m(v_m)$.The quality of such an assignment may be assessed by computing the deformation which occurs in the model when this input vertex is associated with v_m, which is expressed by an updated version of v_d, whose attributes are thus merged with those from v_i. Thus, after the

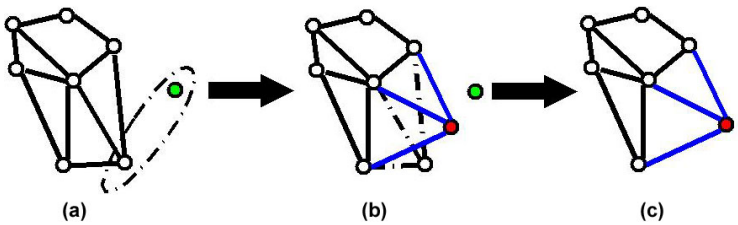


Figure 4: (a) A deformation graph and an input (green) vertex. Initially, each vertex $v_d \in V_d$ is a copy of its corresponding model vertex v_m, and the highlighted pair represents an input vertex v_i and a deformation graph vertex $v_d = v_m$ to be associated. This association represents the fusion of the image regions which are represented by these input and model vertices. (b) The red vertex represents the vertex from the deformation graph whose object attributes were updated with those from the input and model vertices. The blue edges depict the relational attributes which were also updated due to the changes in the attributes of v_d. In order to illustrate the deformation graph computation, let $\mu(v_i) = (10, 20, 30)$ and $\mu(v_m) = (10, 10, 10)$. In this case, $\mu'_d(v_d) = (10, 15, 20)$. Analogously, let $(160, 100)$ and $(100, 200)$ be the centroids of the regions represented by v_i and v_m, respectively. The resulting centroid is represented by the deformation vertex v_d and is the average point between v_i and v_m, at $(130, 150)$. This fact leads to the update of attributes of all edges with an endpoint at v_d. For instance, let v be a neighbor of v_d with centroid at $(90, 80)$, $e_d = (v_d, v) \in E_d$. Then, $\nu'_d(e_d) = (\frac{-40}{d_{max}}, \frac{-70}{d_{max}})$. (c) The resulting deformation graph.

update, $G_d(v_i, v_m)$ becomes a distorted version of the model in which each object attribute (appearance information) $j, j = 1, \ldots, |L_V|$, of the deformed vertex is computed as:

$$\mu'_d(v_d)[j] = \sigma^j_\mu(\mu_d(v_d)[j], \mu_i(v_i)[j]) \tag{1}$$

and each relational attribute $k, k = 1, \ldots, |L_E|$, of the edges connected to the deformed vertex is calculated as:

$$\nu'_d(e_d)[k] = \sigma^k_\nu(\nu_d(e_d)[k]) \tag{2}$$

$\forall e_d \in E_d(v_d) = \{e \in E_d : e = (v_d, w_d) \text{ or } e = (w_d, v_d), w_d \in V_d\}$, i.e. for all edges e_d linked to v_d.

The functions σ^j_μ and σ^k_ν are called *update* functions and they shall be covered in practice in Section 5.1. Thus, the new value of the j-th object attribute vector of v_d after its fusion with v_i is obtained through a function σ^j_μ which expresses the mixture of the "j-th" input object attribute with the "j-th" original model object attribute. For example, if the $j-$th object attribute is the average color of the region represented by a vertex, then $\sigma^j_\mu(\mu_d(v_d)[j], \mu_i(v_i)[j])$ is the average color of the regions represented by v_d and v_i as if they were a single region. Similarly, because v_d was updated with object attributes of v_i, the edges linked to v_d must have their associated relational attributes updated according to functions $\sigma^k_\nu, k = 1, \ldots, |L_E|$, of choice.

The impact of each local deformation due to the association of a pair of vertices $v_i \in V_i$ and $v_m \in V_m$ is measured according to the following *cost*

function:

$$f(G_d, G_m) = \alpha.c_V(v_d, v_m) + \frac{(1-\alpha)}{|E_d(v_d)|}. \sum_{e_d \in E_d(v_d)} c_E(e_d, e_m) \tag{3}$$

This cost function measures how the association of a vertex v_i with a copy v_d of a model vertex affects the local structure of the graph (second term), as well as the appearance attributes it holds (first term). The parameter α, $0 \leq \alpha \leq 1$, controls the importance of the appearance versus the structural attributes.

The term $c_V(v_d, v_m)$ is a measure of the deformation occurred in the object attributes of v_d in comparison with v_m, being defined as:

$$c_V(v_d, v_m) = \sum_{j=1}^{|L_V|} \beta_V^j.\delta_V^j(\mu'_d(v_d)[j], \mu_m(v_m)[j]) \tag{4}$$

whereas $c_E(e_d, e_m)$ is a measure of the deformation implied in the relational attributes of $e_d \in E_d(v_d)$ with respect to its corresponding original model edge $e_m \in E_m$, defined as:

$$c_E(e_d, e_m) = \sum_{k=1}^{|L_E|} \beta_E^k.\delta_E^k(\nu'_d(e_d)[k], \nu_m(e_m)[k]) \tag{5}$$

Both δ_V^j and δ_E^k are *dissimilarity* functions which compare each deformed attribute associated to a vertex or an edge with those from the original model. Thus, the higher the value of the dissimilarity function between attribute vectors is, the higher is the resulting value of $f(G_d, G_m)$. Constants β_V^j and β_E^K balance the importance among the vertex and edge attributes, respectively.

4 The optimization algorithm

The following graph matching algorithm was devised in order to deal with the topological issue using the deformation graph concept. It takes G_i and G_m as input and maps each vertex from G_i to a single vertex in G_m, as detailed in the pseudo-code below. The output of the algorithm is a mapping of all vertices of V_i to vertices of V_m.

---

define α, β_V^j, for $j = 1, \ldots, |L_V|$, and β_E^k, for $k = 1, \ldots, |L_E|$
$G_d \leftarrow G_m$
for each vertex $v_i \in V_i$ **do**
  $f_{min} \leftarrow \infty$
  $minlbl \leftarrow -1$
  for each vertex $v_m \in V_m$ **do**
    let $v_d \in V_d$ be the vertex corresponding to v_m
    let $E_d(v_d)$ be the set of all edges linked to v_d
    save the original attribute vectors $\mu_d(v_d)$ and $\nu_d(e_d)$, $\forall e_d \in E_d(v_d)$
    for each attribute $j = 1, \ldots, |L_V|$ of vertex $v_d \in V_d$ **do**

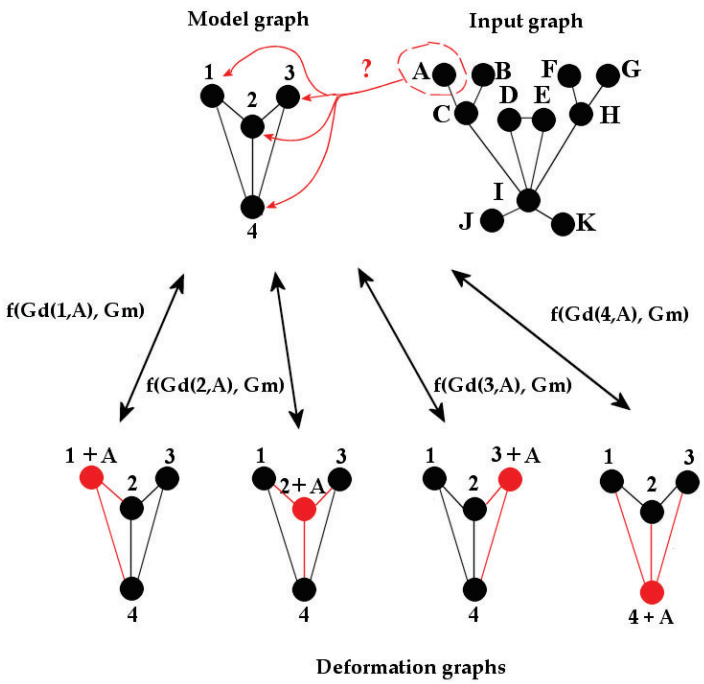


Figure 5: A graphical illustration for the matching of an input vertex to a model vertex according to the implied deformations. Input vertex A must be matched to one of the model vertices $(1 \ldots 4)$. At each iteration, the deformation graph starts as a copy of the model graph and it is then updated for each possible match of A with a model vertex, while the corresponding values of the cost function are computed. The deformations implied by each assignment are highlighted in the picture as red arcs and red vertices, and these distorted vertex and edges are compared to the original ones present in the model. Finally, the match leading to the lowest cost is chosen as the solution for A.

```
        μ'_d(v_d)[j] ← σ_μ(μ_d(v_d)[j], μ_i(v_i)[j])
      end for
      for each e_d ∈ E_d(v_d) do
        for each attribute k = 1, ..., |L_E| of e_d do
          ν'_d(e_d)[k] ← σ^k_ν(ν_d(e_d)[k])
        end for
      end for
      compute the value f of the cost function f(G_d(v_i, v_m), G_m) (Eq. 3)
      if f < f_min then
        f_min ← f
        minlbl ← v_m
      end if
      restore the original attribute vectors of μ_d(v_d) and ν_d(e_d), ∀e_d ∈
      E_d(v_d)
    end for
    label(v_i) ← minlbl
  end for
```

The algorithm works as follows. Firstly, it is necessary to set the parameters α, β_V^j, $j = 1, \ldots, |L_V|$, and β_E^k, $k = 1, \ldots, |L_E|$. These parameters weigh the influence of the appearance against the structural information (parameter α, Eq. 3), as well as the importance of each attribute.

Line 2 initializes the *deformation* graph G_d with a copy of the original model G_m. Line 3 loops through all the vertices $v_i \in V_i$ from the input ARG G_i, so that all of them are mapped to a vertex in G_m. Line 6 loops through all the vertices $v_m \in V_m$ from the model ARG G_m, in order to analyze each candidate pair (v_i, v_m). First, the attributes of the vertex v_d corresponding to v_m in the *deformation* graph are updated, as well as those of the edges linked to v_d (lines 10–17). Then (line 18), the algorithm computes the impact of the deformation expressed by $G_d(v_i, v_m)$ by calculating the corresponding value of the cost function (Eq. 3), while keeping track of the model vertex which led to the cheapeset solution found so far for that v_i (lines 19–22). Finally, v_i is mapped (labelled) to the model vertex which is associated to the cheapest solution (line 25). This process is repeated until all input vertices are evaluated and it results in each input vertex labelled with a single model vertex.

We shall briefly analyze the algorithm complexity. For each association of vertices from the input and model graphs, the deformation graph is recomputed. This implies constant-time computations for the update of an object attribute vector and, at most $|E_m|$ constant-time updates of relational attributes. The same holds true for the evaluation of the cost function value (Eq. 3). Thus, the algorithm is $\Theta(|V_i|.|E_m|)$, which also means that the running time varies according to the adopted model graph topology. The worst-case happens when the model graph is actually complete. Nevertheless, simpler topologies, such as that of a planar graph, require significantly less time and produce successful segmentations, as Section 5 shows.

Fig. 5 illustrates how the algorithm works using the *deformation* graph. In this example, an input vertex A shall be matched to one of the four model vertices (1 through 4). Thus, all possible assignments are evaluated and the cost function is computed for each possible match. The final label of A corresponds to the model vertex which was least affected by the deformation introduced by the attributes of v_i.

5 Experiments

In order to test the present technique, a Java application was designed and implemented [1]. Its interface allows the user to load images to be segmented, create a model image according to traces drawn over different regions of interest of the input image, and define the parameters of the cost and dissimilarity functions (Eqs. 3, 8 and 9).

Tests have been performed using sample natural static images from the Berkeley Image Segmentation Database [2] and frames of two sample video

[1] Visit `http://structuralsegm.sourceforge.net/` for more results and demo videos.

[2] `http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/`

sequences. Model images have been created with the aid of the interface in order to point out the objects of interest for the purpose of segmentation.

In all experiments, the input ARGs derived for each input image present the same topology: they include a vertex for each region produced by the oversegmentation and there are edges connecting two vertices representing adjacent regions. On the other hand, the model ARGs followed either a triangulated topology [3] (Fig. 7 right) or a complete graph structure. These different levels of structural information embedded in the model graphs allow the assessment of how segmentation quality and computer performance vary in such cases.

A set of object and relational attributes, as well as update and dissimilarity functions, have been systematically used in all performed tests. However, the weight parameters of the cost and dissimilarity functions are not necessarily the same, since they have been adopted according to the characteristics of each image under analysis. All these shall be described in the remainder of this section.

5.1 Attributes

For color images, the object attribute vector was composed of a single attribute ($|L_V| = 1$): the average $L^*a^*b^*$ tuple characterizing the corresponding image region, i.e. $\mu(v) = (L^*_v, a^*_v, b^*_v)$. The CIELAB [31] color space was chosen because it is a closer representation of the way humans percept color and its metrics were suitable and straightforward for the idea of mixing attributes from two image regions when building the deformation graph, as explained in the next subsection. When dealing with gray-scale images, $\mu(v) = (g(v))$, where $g(v)$ denotes the average gray-level of the image region associated to vertex $v \in V$. Each $L^*a^*b^*$ component or gray-scale value is normalized between 0 and 1 with respect to the minimum and maximum possible range values.

Similarly, a single relational attribute ($|L_E| = 1$) is associated to each edge $e = (v, w) \in E$, $v, w \in V$, being defined as $\nu(e) = \frac{(p_w - p_v)}{d_{max}}$, where p_v and p_w are the centroids of their corresponding image regions. The constant d_{max} represents the largest distance between any two points of the input image region and it is used to keep the modulus of each vector between 0 and 1. Thus, the relational attribute is a geometric vector normalized by d_{max}, whose origin is the centroid of the source vertex and whose tip is the centroid of the sink vertex. Other attributes may easily be employed, since the methodology presented herein does not impose any restriction on the nature of μ and ν.

5.2 Update functions

When searching for the best correspondence between an input and a model vertex, different deformation graphs are obtained. As explained in Section 3,

[3] In the segmentation problem, each vertex stores the centroid of its respective region, and triangulated edges were obtained by a Delaunay Triangulation [30] based on these centroids, since it induces a planar graph and maintains the connectivity between adjacent regions.

each G_d expresses the fusion of a given input vertex v_i and a vertex v_d. Thus, the object and relational attribute vectors of v_d must be updated after each fusion, in order to reflect the deformation under analysis.

The following updating functions have been devised to update the aforementioned attributes:

$$\mu'_d(v_d) = \sigma_\mu(\mu_d(v_d), \mu_i(v_i)) = (\frac{L^*_{v_d} + L^*_{v_i}}{2}, \frac{a^*_{v_d} + a^*_{v_i}}{2}, \frac{b^*_{v_d} + b^*_{v_i}}{2}) \quad (6)$$

$$\nu'_d(e_d) = \sigma_\nu(\nu_d(e_d)) = \frac{(p_{w_d} - p'_{v_d})}{d_{max}} \quad (7)$$

$\forall e_d = (v_d, w_d) \in E_d(v_d)$, $p'_{v_d} = \frac{p_{v_d}+p_{v_i}}{2}$, $p_{v_d} = p_{v_m}$ before the update.

Thus, the new value of the object attribute $\mu'_d(v_d)$ after merging the attributes of v_i and v_d (or equivalently, v_m) expresses the average color of all pixels of the regions represented by v_i and v_m, as if they were a single region. The structural impact after associating v_i and v_m is represented in G_d by the updated relational attribute vector ν'_d of each directed edge linked to v_d. This update expresses the distorted geometric vectors which connect the points corresponding to the centroid of v_d, which is the average of the centroids of v_i and v_m, and the centroids of its neighbor vertices.

5.3 Dissimilarity functions

Once a deformation graph G_d is computed to express the distortion associated to a given matching candidate pair, the respective value of the cost function f (Eq. 3) must be computed in order to measure the level of dissimilarity between G_d and the original model G_m. Since f is defined in terms of vertex and edge dissimilarity costs, it is necessary to define distance metrics for object and relational attributes (appearance and structure, respectively).

To compute the deformation between the object attributes of v_d and v_m, the term $c_V(v_d, v_m)$ is defined as:

$$c_V(v_d, v_m) = \beta_V.\delta_V(\mu'_d(v_d), \mu_m(v_m)) = \sqrt{\sum_{C=L^*,a^*,b^*} (C_{v_d} - C_{v_m})^2} \quad (8)$$

Similarly, if $e_d \in E_d(v_d)$ and $e_m \in E_m$ is its corresponding model edge, then $c_E(e_d, e_m)$ is a measure of the deformation between both edges defined as:

$$c_E(e_d, e_m) = \beta_E.\delta_E(\nu'_d(e_d), \nu_m(e_m)) = \gamma_E.\frac{|\cos(\theta) - 1|}{2} + (1-\gamma_E).|\|\nu(e_d)\| - \|\nu(e_m)\||$$
$$(9)$$

The value θ is the angle between the vectors $\nu(e_d)$ and $\nu(e_m)$, whereas the parameter γ_E, $0 \leq \gamma_E \leq 1$, controls the weights of the angular and modular dissimilarities between vectors. The angular cost (Fig. 6) is high for opposite vectors and low for vectors with similar orientation. Thus, the structural distance (the second term in Eq. 3) is computed as the average over all distorted edges e_d (directly connected to v_d) compared to their respective model edges $v_m \in V_m$. Note that, because a single object or relational attribute is being considered in each dissimilarity function, the weights β_V and β_E are both set to 1.

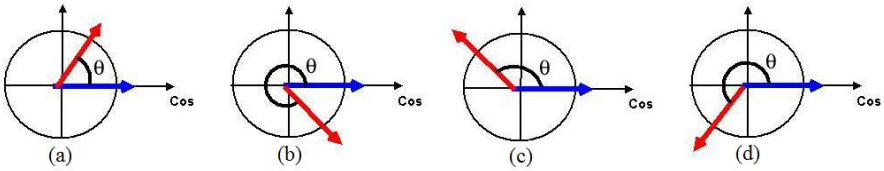


Figure 6: The angular cost in Eq. 9 results in lower values when the vectors have similar orientations (a, b) and higher values for opposite orientations (c, d).

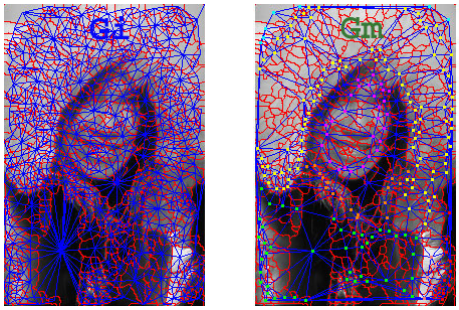


Figure 7: Input and model graphs: based upon the watershed regions, both input and model graphs are created. Edges are depicted by blue lines. Input edges (E_i) represent the adjacency between watershed regions and model edges (E_m) represent the adjacency of regions of a Delaunay triangulation obtained from the centroids of the regions which intercept the user scribbles, as highlighted in the image on the right.

5.4 Results on static images

The experiments presented herein depict several segmentations of sample natural gray-scale and color images obtained from the Berkeley Image Segmentation Database. The test environment comprised a machine with a 1.4MHz CPU and 1.5Gb RAM.

Each picture was segmented according to the following protocol. First, the user scribbles over the input image, using a different color to identify each object or part to be segmented. Next, the input image is oversegmented through the watershed algorithm, giving rise to the input and model graphs (Fig. 7) as described in Section 2.2. The topology used in these examples was that of a triangulated graph, in which the edges are obtained by a Delaunay triangulation [30] from the centroids of the regions represented by the vertices. Then, the α and γ parameters are set and the matching algorithm proposed herein is applied, leading to the object segmentation.

Sample resulting images (Figs. 8 and 9) show the final regions labelled according to the color of the strokes defined by the user. Although certain image regions, such as the geisha, the mountains, and the eskimo, present high variability due to textures or varying appearance, the final segmentation remains accurate and robust thanks to the structural constraints embedded in the model.

Another set of experiments aimed at evaluating the performance of the algorithm in terms of result quality and speed, when distinct topologies have been carried out. For this purpose, images were segmented using either a

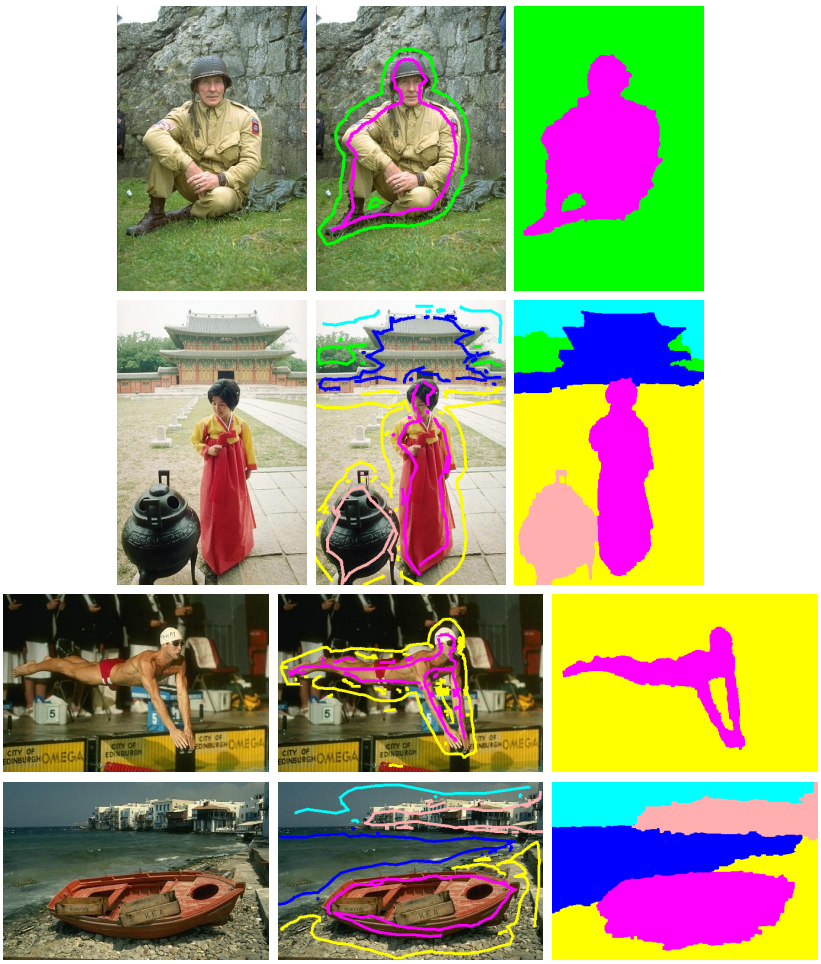


Figure 8: Sample color images from the Berkeley dataset used to test the proposed method: 'soldier', 'geisha', 'swimmer' and 'boat'. From left to right: input image, scribbles drawn by the user over the input image and the final segmentation.

triangulated model graph or a complete graph, in which there is an edge connecting every pair of vertices. Simultaneously, these experiments studied the influence of appearance versus structural information (Eq. 3), since segmentations were obtained using $\alpha = 0.0, \ldots, 1.0$, keeping $\gamma_E = 0.5$ in order to assign the same importance to the angular and modular costs (Eq. 9). Figs. 10 through 12 present sample results of this experiment.

Based upon the obtained results, it can be seen that, as α increases, the importance of the appearance information increases, resulting in more fragmented regions. Thus, the variation of α reflected the influence of the appearance versus the structural information over the segmentation, producing a better result for α between 0.1 and 0.4, for complete graphs, and between 0.1 and 0.6, for triangulated models. In terms of speed, because the algorithm runs in time proportional to the number of model edges, the running times for the segmentations using complete graphs were, as expected, much higher than those obtained for triangulated models, as shown in Tables 1 and 2.

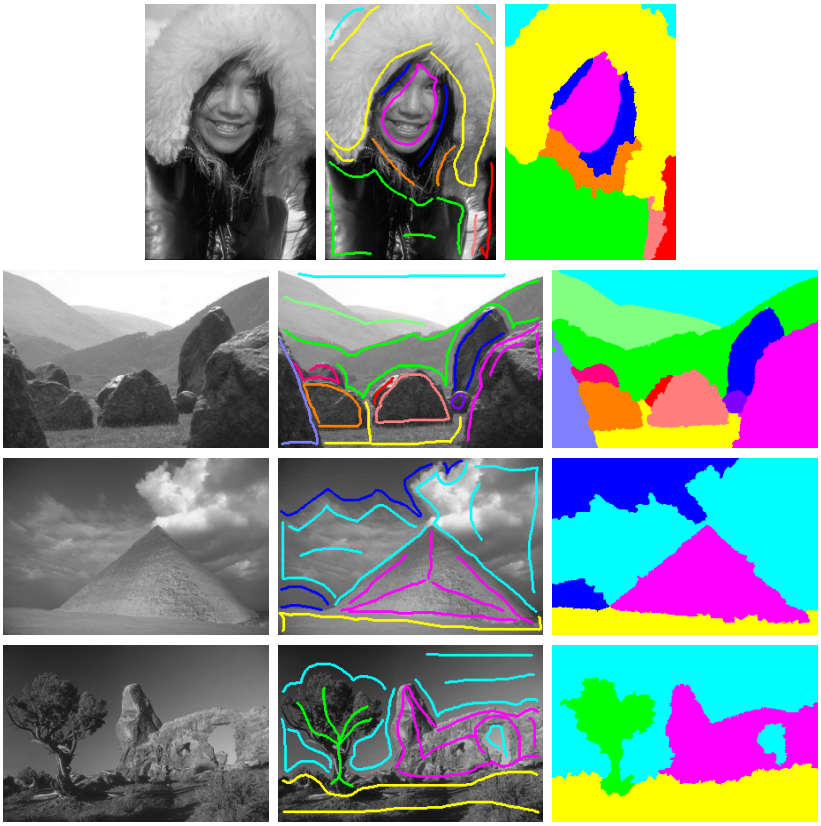


Figure 9: Sample gray-scale images from the Berkeley dataset used to test the proposed method: 'eskimo', 'rocks', 'pyramid' and 'landscape'. From left to right: input image, scribbles drawn by the user over the input image and the final segmentation.

Figure 10: Results from the proposed algorithm using a complete model graph, for $\alpha = 0.0, \ldots, 1.0$.

Moreover, the deformation graph using triangulated models better preserved the uniform regions in the segmentation (as α increases) than using complete model graphs. This suggests that the triangulated models are

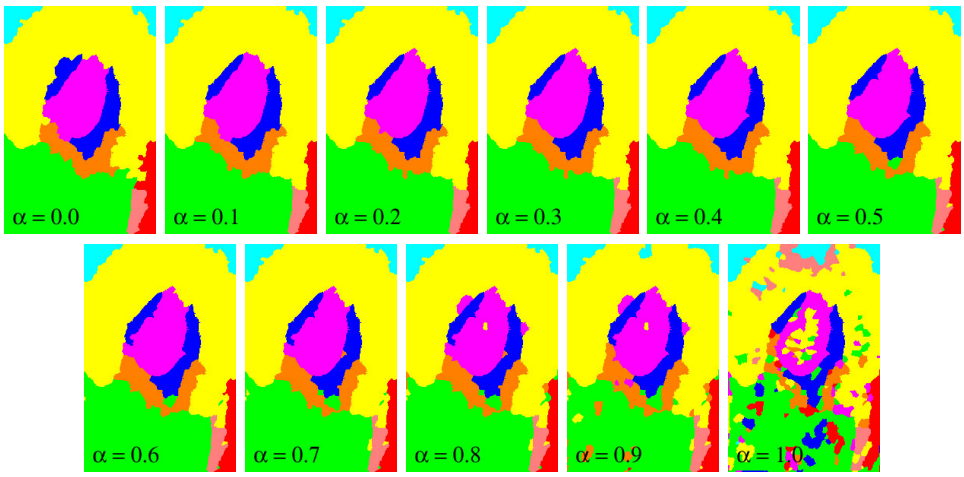


Figure 11: Results using a triangulated model graph, for $\alpha = 0.0, \ldots, 1.0$. Note that these results present more uniform regions than the ones obtained using a complete model (Fig. 10).

Segmented Image	$\lvert V_i \rvert$	$\lvert E_i \rvert$	$\lvert V_m \rvert$	$\lvert {E_m}^C \rvert$	$\lvert {E_m}^T \rvert$
eskimo	546	3050	224	49952	1308
rocks	841	4802	313	97656	1844
pyramids	822	4642	327	106602	1928
landscape	998	5658	367	134322	2170

Table 1: The sizes of the input and model graphs used in the experiments of performance evaluation and comparison between the proposed algorithm DG and the SFS method, in which $\lvert {E_m}^C \rvert$ and $\lvert {E_m}^T \rvert$ denote the number of edges in the complete and in the triangulated model graphs, respectively.

more adequate to represent structural information, since each region considers only the structural relations from its immediate neighbors, providing more relevant context information than farther regions.

Finally, the present algorithm was compared to the SFS - sequential forward search - matching algorithm described in [26]. In summary, this algorithm takes as input two graphs (the input and the model) and finds a mapping between their set of vertices. First, the algorithm computes, for each input graph vertex, its corresponding *supervertex*, an auxiliary structure that represents an ordered list of all possible mappings between that given input vertex and all model ones. This order is defined according to the increasing Euclidean distances between the input and model vertex centroid attributes and the closest pair defines the cost of its respective supervertex. Then, the set of supervertices is also sorted in increasing order of costs, defining an order in which input vertices should be visited to determine the total mapping. Each input vertex v_i is analysed in this particular order and the algorithm computes the cost of the whole solution obtained so far. However, the cost of the solution only gradually takes structure into account. Also, the algorithm copes with differences in the topologies of both graphs by evaluating four distinct possible configurations and by computing edge dissimilarities accordingly. Nevertheless, the considered configurations

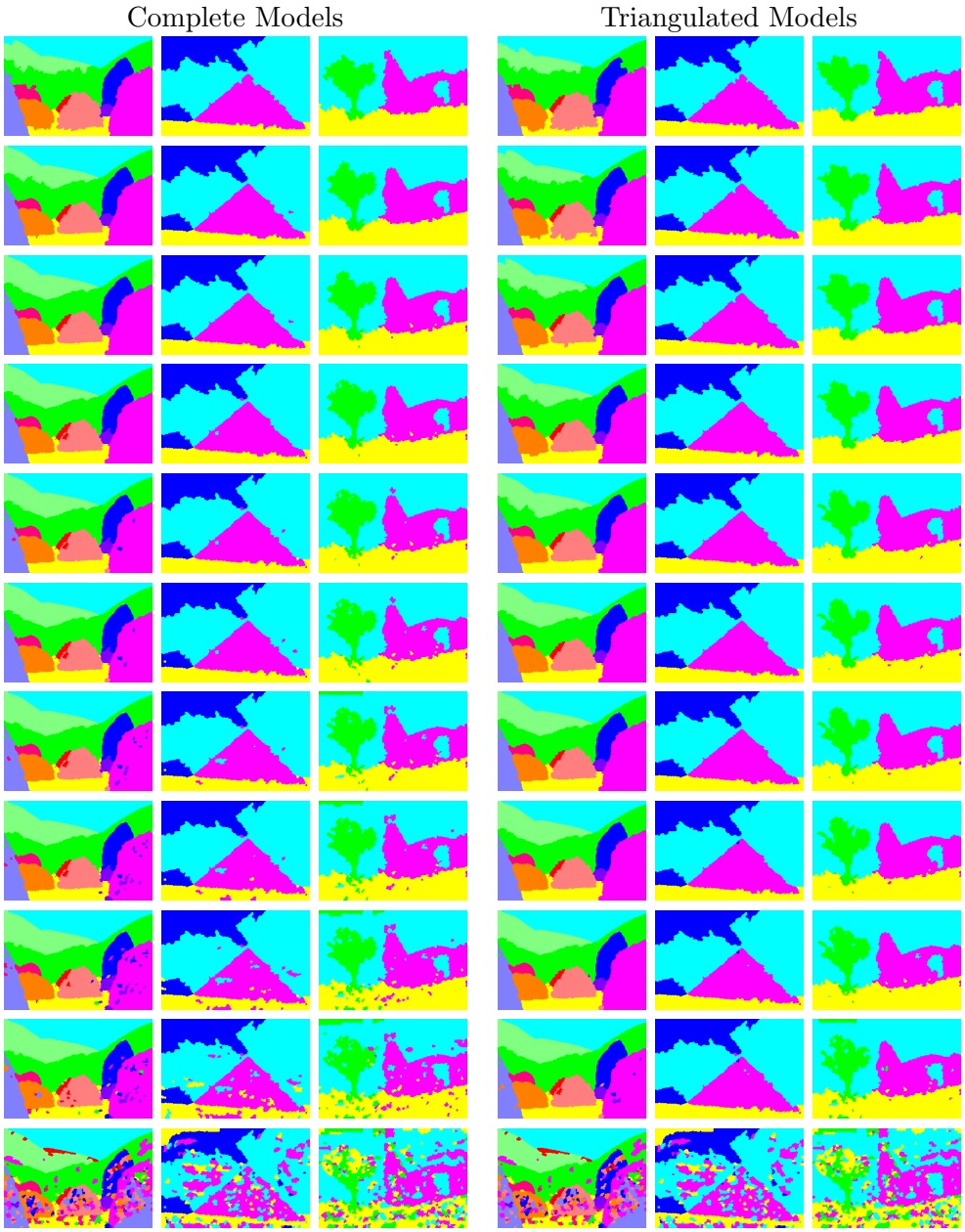


Figure 12: Results using complete or triangulated model graphs, for $\alpha = 0.0, \ldots, 1.0$. Each line presents the results for a given α for all three images, in increasing order from top to bottom, using complete and triangulated model graphs.

do not cover the whole range of structural possibilities and may impair the final segmentation.

The comparison between both algorithms was achieved through tests performed over the same set of gray-scale images and using the same user scribbles for each image. Both algorithms were implemented under the same Java framework and they have been presented with the same pair of input and model graphs for each segmentation. As in the previous experiment, two types of graphs were used - triangulated and complete - and the α parameter

Segmented Image	Complete model		Triangulated model	
	DG	SFS	DG	SFS
eskimo	8.8	10.8	1.5	3.9
rocks	25.5	35.6	3.2	13.4
pyramids	27.7	35.9	3.2	17.6
landscape	40.5	59.2	4.3	27.7

Table 2: The average times (in seconds) for segmenting the given gray-scale images using either the matching procedure proposed in this paper (DG) or the SFS algorithm from [26]. Complete and triangulated models were used and both algorithms were implemented in Java and ran under the same conditions for the purpose of comparison.

was varied. Using complete model graphs, both algorithms produced similar results and responded alike to the variation of α. On the other hand, the use of triangulated model graphs shows that the present method outperforms the SFS algorithm in terms of segmentation accuracy.

Note that the bad segmentation produced by the SFS when $\alpha = 0.0$, i.e. only structural information is considered, is due to the fact that the small number of edges in the triangulated model impairs the accurate comparison between input and model edges used to compute the edge dissimilarity. Thus, due to the order given by the supervertices, the first input vertices classified by the SFS correspond to the input regions in the model, and they are not guaranteed to be associated to the correct model region, compromising the whole structure information embedded in the solution. On the other hand, the present matching algorithm does not depend on the order in which vertices from the input are labelled, since each vertex is treated separately as a deformation of the model.

The new algorithm proposed for the graph matching step, using triangulated model graphs, presents time complexity $O(|V_i|.|E_m|)$, with $|E_m| = O(|V_m|)$, for triangulated model graphs, and $|E_m| = O(|V_m|^2)$, for complete model graphs. The SFS algorithm, however, is bounded by a function $\Omega(|V_i||V_m|^2)$. Thus, even though they share the same upper bound for complete model graphs, the running times of the methods were not alike, with the SFS algorithm taking longer for all inputs. On the other hand, the use of triangulated model graphs decreases the time complexity by a factor of $|V_m|$, which drastically drops the time consumed by the present algorithm. Tables 1 and 2 highlight these differences in speed for the set of tests performed, presenting their approximate average running times.

5.5 Model robustness and reuse

To reduce user interaction, experiments have been performed in order to evaluate the robustness and reusability of the model when applied to different images presenting similar objects. The set of input images to be segmented comprised sample frames of a moving head video from the XM2VTS Database [4] and a set of random motorcycle images (Fig. 14). For these tests, the adopted model graph topology was that of triangulated graphs,

[4] `http://www.ee.surrey.ac.uk/CVSSP/xm2vtsdb/`

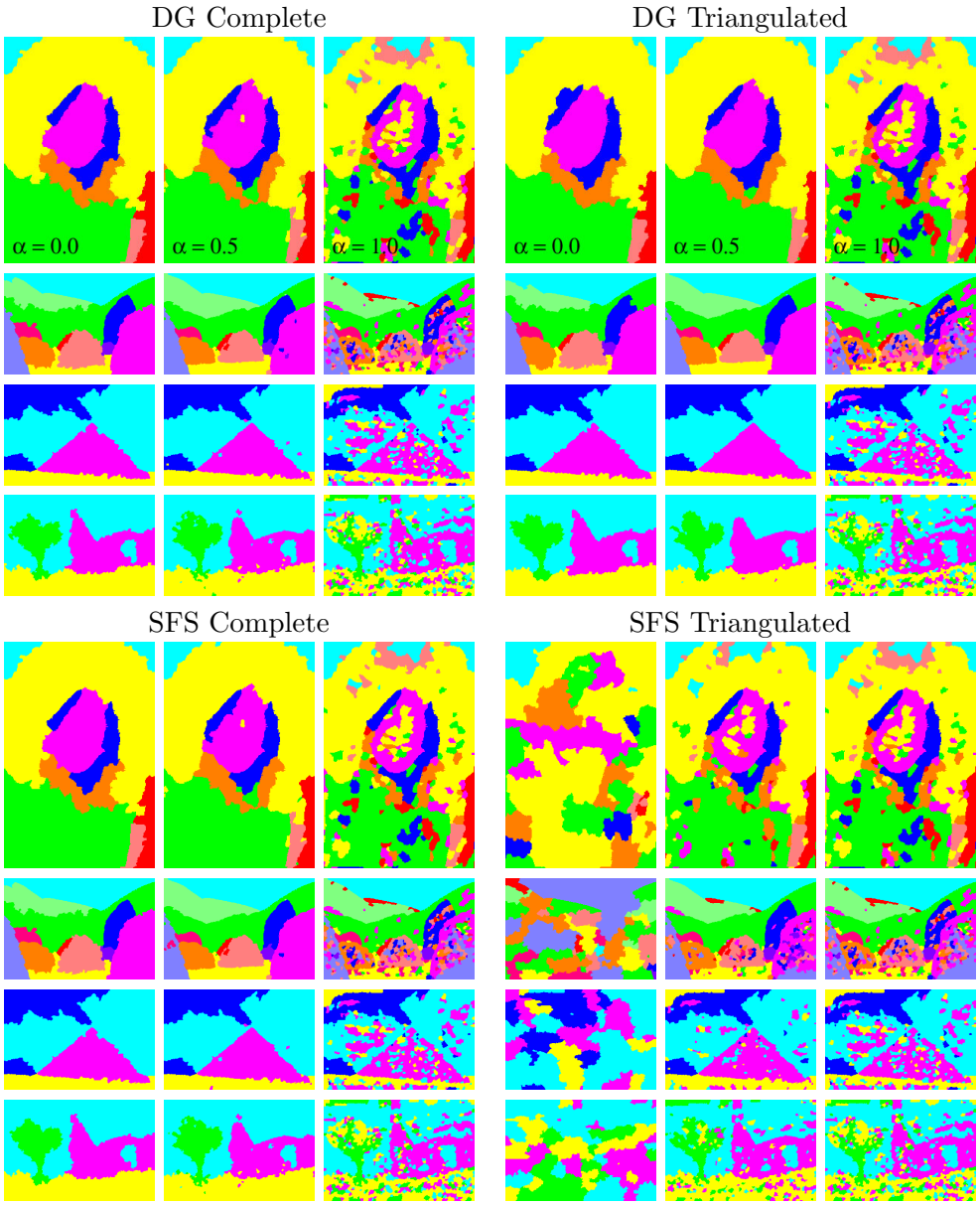


Figure 13: Sample segmentation results of the 'eskimo', 'rocks', 'mountains', and 'landscape' images obtained from both algorithms (the present method DG and the SFS) using complete and triangulated model graphs, for $\alpha = 0.0, 0.5$, and 1.0.

since they have proven to produce good results in less time, as previously discussed.

The first test is depicted in Fig. 15 and it works as follows. The first object, i.e. the woman's head subdivided into parts, is interactively segmented in the same manner as a single static image (Section 5.4). Thus, a triangulated model graph is obtained from the intersection of the user scribbles with the watershed regions that fall inside the subimage defined by the minimum rectangle that encloses the scribbles. The input graph is

Figure 14: Input images with multiple similar objects in order to test the reusability of the triangulated models.

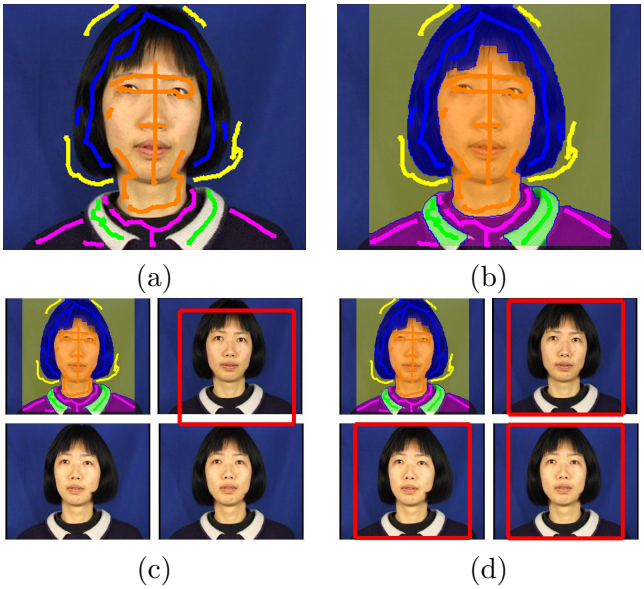
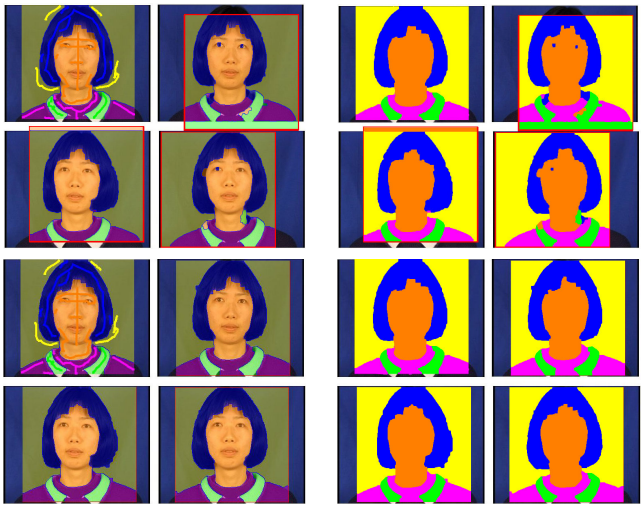


Figure 15: Reusing one single triangulated model in order to segment similar objects. First, the user performs a regular segmentation in one object by defining the scribbles (a) to segment it (b). Then the user places the red rectangle (c) over each desired object to be segmented, as indicated in (d). The same model created in the first segmentation is used to segment all other similar objects, thus reducing the level of user interaction.

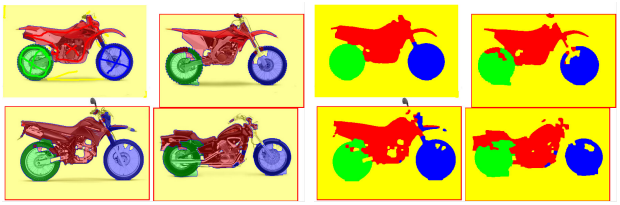
obtained similarly, but from all the watershed regions inside the rectangle. Then, the other segmentations are obtained by placing the rectangle over the other objects, obtaining new input graphs, and computing the segmentations through the matching between each input graph and the previously obtained model. Despite the fact that the deformation graph depends on the centroids of the regions in order to calculate the graph matching, the method is robust to small displacements when the appearance information compensates the structural 'noise' (Fig. 16 (a), bottom left result).

A similar test was applied to the motorcycles in Fig. 17. As the figure shows, the results are sensitive to the rectangle position since the appearance changes from one motorcycle to another. However, the structures are similar and, thus, for better results, greater influence was given to the structure ($\alpha = 0.1$). Because of this, the worst segmentation was that of the last motorcycle (on the bottom right), since its structure differs from the other

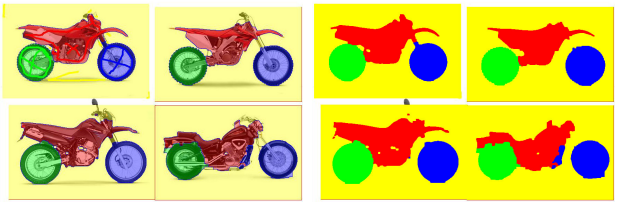


(a) (b)

Figure 16: (a) A distorted placement of the rectangles. (b) The 'optimum' placement of the rectangles. In both cases, it was used the proposed method with $\alpha = 0.4$ in order to obtain the matching.



(a)



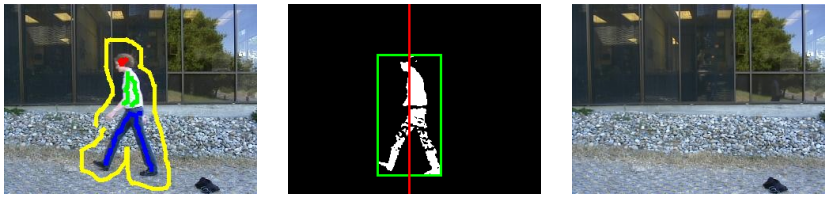
(b)

Figure 17: (a) A distorted placement of the rectangles ($\alpha = 0.4$). (b) The 'optimum' placement of the rectangles ($\alpha = 0.1$).

motorcycles (e.g. larger distance between wheels).

Finally, the reusability of the model was also tested on the digital video sequence "Person walking straight" [5] presented in [32]. To obtain the triangulated model, user scribbles were drawn over one particular frame selected from the sequence (Fig. 18 (a)) and the proposed algorithm was applied using this single model in order to segment the rest of the sequence. The additional information of the X axis position of the person was also used to update the centroid coordinates of each part represented by a model vertex.

[5] `http://www.csc.kth.se/~hedvig/data.html`



(a) (b) (c)

Figure 18: (a) Selected frame and the user traces for the triangulated model. (b) Position calculated through background subtraction: first, the object is detected by background subtraction with threshold; then, the position of the object in the X axis is given by the red line dividing in the middle the minimum (green) rectangle that encloses the thresholded components. (c) The median pixels of the first five frames used as the background model.

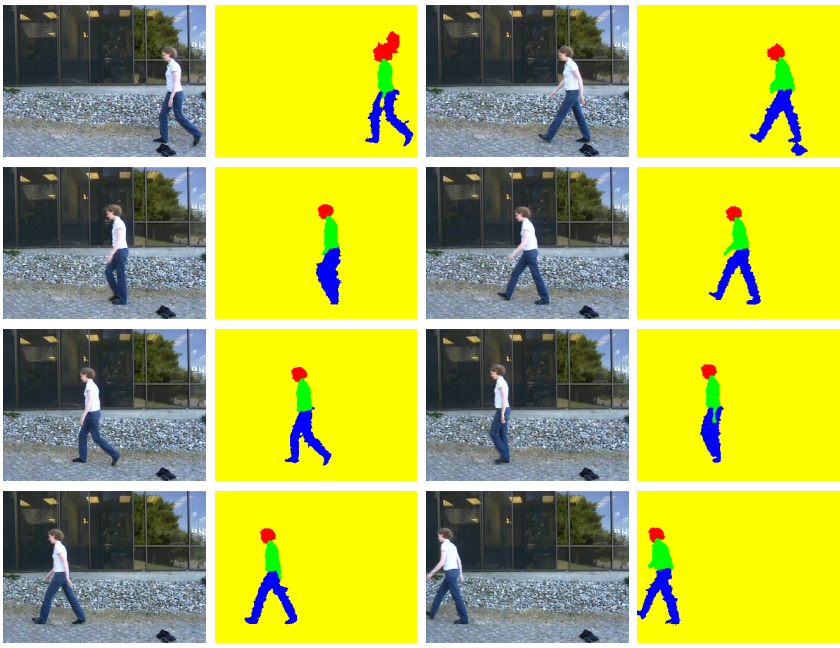


Figure 19: The results of the proposed method for a digital video sequence using the triangulated model in Fig. 18 (a).

This information was obtained by a simple background subtraction (Fig. 18 (b)), which used the median pixels of the first five frames as the background model (Fig. 18 (c)). The input graphs were created from the watersheds of each frame. Results are illustrated in Fig. 19.

6 Conclusion

This paper proposed a novel algorithm for performing model-based segmentation and recognition of object parts in images using attributed relational graphs to represent both input and model. This approach takes into account both appearance and structural (the spatial relations between the vertices) information in order to obtain a matching between input and model graphs.

Topological differences between both graphs are dealt with by means of a deformation ARG, a structure which evaluates the local impacts (or deformations) caused by the association of a given input vertex with a model vertex. Since structure is at stake, two graph topologies have been evaluated: triangulated model graphs and complete model graphs. The new algorithm using triangulated models produced successful segmentation results with fast performance. Also, the method has presented model robustness when recognizing object parts based on appearance and structural information, as well as model reusability when segmenting similar objects in several images.

Our ongoing work is devoted to improving model robustness. For instance, an algorithm robust to translation is desirable in order to allow the segmentation of multiple objects without placing a rectangle over each object and to recognize the object parts in a video sequence without providing the information about the position of the object. This shall be accomplished through an investigation of a MAP-MRF model within this framework.

Acknowledgements. This work was supported by FAPESP, CAPES, CNPq, COFECUB and FINEP.

References

[1] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition," *Int. J. Comput. Vision*, vol. 61, no. 1, pp. 55–79, 2005.

[2] A. Mohan, C. Papageorgiou, and T. Poggio, "Example-based object detection in images by components," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 4, pp. 349–361, 2001.

[3] R. M. Cesar-Jr., E. Bengoetxea, I. Bloch, and P. Larrañaga, "Inexact graph matching for model-based recognition: Evaluation and comparison of optimization algorithms," *Pattern Recognition*, vol. 38, no. 11, pp. 2099–2113, 2005.

[4] D. Crandall, P. Felzenszwalb, and D. Huttenlocher, "Spatial priors for part-based recognition using statistical models," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, vol. 1, 2005, pp. 10–17.

[5] A. X. Falcão, J. Stolfi, and R. A. Lotufo, "The image foresting transform: Theory, algorithms, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 1, pp. 19–29, Jan 2004.

[6] A. Blake, C. Rother, M. Brown, P. Pérez, and P. H. S. Torr, "Interactive image segmentation using an adaptive GMMRF model," in *Proc. European Conf. Comput. Vision*, 2004, pp. 428–441.

[7] C. Rother, V. Kolmogorov, and A. Blake, ""GrabCut": interactive foreground extraction using iterated graph cuts," in *ACM SIGGRAPH*, 2004, pp. 309–314.

[8] A. K. Sinop and L. Grady, "A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm," in *Proc. Int. Conf. Comput. Vision*, 2007, pp. 1–8.

[9] Y. Boykov and G. Funka-Lea, "Graph cuts and efficient n-d image segmentation," *Int. J. Comput. Vision*, vol. 70, no. 2, pp. 109–131, 2006.

[10] Y. Boykov, O. Veksler, and R. Zabih, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 9, pp. 1124–1137, 2004.

[11] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis, "The complexity of multiway cuts," *ACM Symp. Theory of Computing*, pp. 241–251, 1992.

[12] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, 2001.

[13] L. Grady, "Random walks for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 11, 2006.

[14] D. Conte, P. Foggia, C. Sansone, and M. Vento, "Thirty years of graph matching in pattern recognition." *Int. J. Pattern Recognition and Artificial Intell.*, vol. 18, no. 3, pp. 265–298, 2004.

[15] H. Bunke, "Recent developments in graph matching." in *Int. Conf. Pattern Recognition*, 2000, pp. 2117–2124.

[16] H. Bunke and B. T. Messmer, "Recent advances in graph matching," *Int. J. Pattern Recognition and Artificial Intell.*, vol. 11, no. 1, pp. 169–203, 1997.

[17] H. Bunke, "Error correcting graph matching: On the influence of the underlying cost function," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 9, pp. 917–922, 1999.

[18] R. C. Wilson and E. R. Hancock, "Structural matching by discrete relaxation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 6, pp. 634–648, 1997.

[19] M. Gori, M. Maggini, and L. Sarti, "Exact and approximate graph matching using random walks," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1100–1111, 2005.

[20] B. T. Messmer and H. Bunke, "Efficient subgraph isomorphism detection: A decomposition approach," *IEEE Trans. Knowledge and Data Eng.*, vol. 12, no. 2, pp. 307–323, 2000.

[21] A. Cross, R. Wilson, and E. Hancock, "Inexact graph matching using genetic search," *Pattern Recognition*, vol. 30, no. 6, pp. 953–970, 1997.

[22] A. Cross and E. Hancock, "Graph matching with a dual step EM algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1236–1253, 1998.

[23] L. Shapiro and R. Haralick, "Structural descriptions and inexact matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 3, no. 5, pp. 504–519, 1981.

[24] S. Medasani and R. Choi, "Graph matching by relaxation of fuzzy assignments," *IEEE Trans. Fuzzy Syst.*, vol. 9, no. 1, pp. 173–182, 2001.

[25] T. Caelli and S. Kosinov, "Inexact graph matching using eigen-subspace projection clustering," *Int. J. Pattern Recognition and Artificial Intell.*, vol. 18, no. 3, pp. 329–354, 2004.

[26] L. A. Consularo, R. M. Cesar-Jr, and I. Bloch, "Structural image segmentation with interactive model generation," in *Proc. IEEE Int. Conf. Image Processing*, vol. 6, San Antonio, Texas, USA, September 2007, pp. 45–48.

[27] A. Jain and D. Zongker, "Feature selection: Evaluation, application, and small sample performance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 2, pp. 153–158, 1997.

[28] D. Kim, I. Yun, and S. Lee, "A new shape decomposition scheme for graph-based representation," *Pattern Recognition*, vol. 38, no. 5, pp. 673–689, May 2005.

[29] L. Vincent and P. Soille, "Watersheds in digital spaces: An efficient algorithm based on immersion simulations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 6, pp. 583–598, 1991.

[30] L. Guibas and J. Stolfi, "Primitives for the manipulation of general subdivisions and the computation of voronoi diagrams," in *Proc. 15th Annual ACM Symp. Theory of Computing*, 1983, pp. 221–234.

[31] G. Hoffmann, "Cielab colorspace," http://www.fho-emden.de/~hoffmann/cielab03022003.pdf.

[32] H. Sidenbladh, M. J. Black, and D. J. Fleet, "Stochastic tracking of 3d human figures using 2d image motion," in *Proc. European Conf. Comput. Vision*, vol. 2, 2000, pp. 702–718.

Pattern Recognition 45 (2012) 1159–1179

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Interactive image segmentation by matching attributed relational graphs

Alexandre Noma [a,*], Ana B.V. Graciano [a], Roberto M. Cesar Jr [a], Luis A. Consularo [b], Isabelle Bloch [c]

[a] Institute of Mathematics and Statistics, University of São Paulo Rua do Matão, 1010, CEP 05508-090 São Paulo, Brazil
[b] Tribunal Superior Eleitoral, Praça dos Tribunais Superiores—Bloco C—SAS—CEP 70096-900, Brasília-DF, Brazil
[c] Telecom ParisTech, CNRS LTCI, 46 rue Barrault, 75013 Paris, France

ARTICLE INFO

Article history:
Received 27 October 2010
Received in revised form
10 August 2011
Accepted 13 August 2011
Available online 1 September 2011

Keywords:
Interactive image segmentation
Matching attributed relational graphs
Deformed graph
Spatial configuration

ABSTRACT

A model-based graph matching approach is proposed for interactive image segmentation. It starts from an over-segmentation of the input image, exploiting color and spatial information among regions to propagate the labels from the regions marked by the user-provided seeds to the entire image. The region merging procedure is performed by matching two graphs: the input graph, representing the entire image; and the model graph, representing only the marked regions. The optimization is based on discrete search using deformed graphs to efficiently evaluate the spatial information. Note that by using a model-based approach, different interactive segmentation problems can be tackled: binary and multi-label segmentation of single images as well as of multiple similar images. Successful results for all these cases are presented, in addition to a comparison between our binary segmentation results and those obtained with state-of-the-art approaches. An implementation is available at http://structuralsegm.sourceforge.net/.

© 2011 Elsevier Ltd. All rights reserved.

1. Introduction

The goal of the image segmentation problem is to divide the input image into different meaningful regions, e.g. to extract the object of interest from the original background. User intervention is often required, encoding prior information into the process through seeds, such as scribbles that the user draws over an image to identify the regions of interest. Ideally, after roughly placing the scribbles, the remainder of the image is automatically segmented. In practice, the user should be allowed to add/remove scribbles to make corrections on the segmentation process and achieve the desired result. The system must be easy to use and the segmentation algorithm should be fast enough to interact with the user, producing accurate results with minimal effort.

A non-interactive model-based image segmentation method has been introduced in [10], where models were obtained from pre-selected images with their respective manual segmentations. The goal was to propagate these manual segmentations to other similar images, by matching model regions with those from the input images (see Fig. 1), for segmentation and recognition of image structures.

The present paper focuses on interactive image segmentation, where the user provides scribbles which are used as input to segment the image (either the same, or a set of similar ones). Therefore, it is natural to think about how to represent the information provided by the user through the scribbles. Most traditional approaches represent them as markers (watershed), seeds (region growing methods) or graph nodes (graph-cut-like methods). However, we are interested in explicitly representing both the image information (grey-level, color, texture) and the structural information (spatial relations) provided by the scribbles. This is useful in a number of situations where the structure of the scene is important to recognize individual objects. For instance, for disambiguating objects having similar appearance in an image, their spatial arrangement is very useful. In this sense, graphs are a natural choice, since we may represent both types of information in a single structure. Inspired by the model-based approach described in [10], the proposed framework is based on over-segmentation, e.g. produced by the watershed transform [34], to build both input and model graphs. Note that, differently from [10], the proposed method follows an interactive model generation approach. In order to be more user-friendly, instead of using manual segmentations for the models, the new method takes advantage of the scribbles provided by the user in order to propagate the segmentations, in which the model graphs can be interactively updated according to the changes on the scribbles. Moreover, the introduced approach allows segmenting both cases, single and multiple similar images, by using model and input graphs encoding information from the same image or from two different images.

Following a preliminary version of this work, described in [11], we focus on the usual version of the interactive image

* Corresponding author. Tel.: +55 11 5016 1873; fax: +55 11 3091 6134.
E-mail addresses: alex.noma@gmail.com (A. Noma), abvg@vision.ime.usp.br (A.B.V. Graciano), roberto.cesar@vision.ime.usp.br (R.M. Cesar Jr), luis.consularo@tse.jus.br (L.A. Consularo), isabelle.bloch@telecom-paristech.fr (I. Bloch).

0031-3203/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2011.08.017

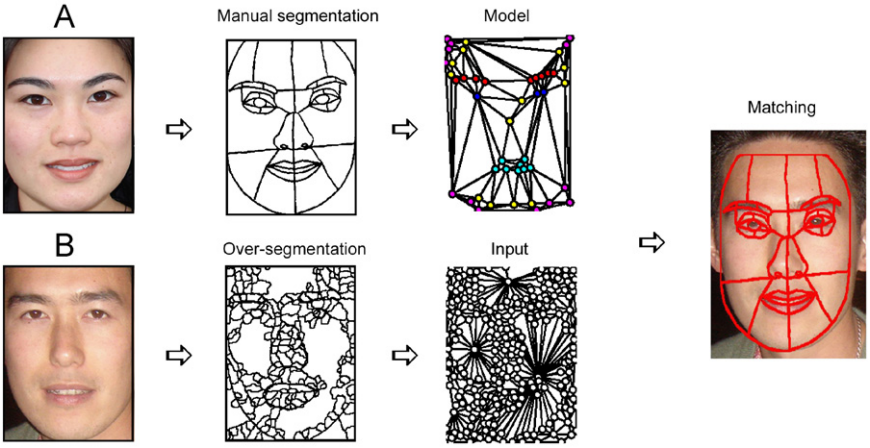


Fig. 1. Our method was inspired by the graph matching approach described in [10]. In that paper, a model graph represented a manual segmentation, while an input graph represented an over-segmentation of another image. The goal was to propagate manual information to other input images for segmentation and recognition of image structures.

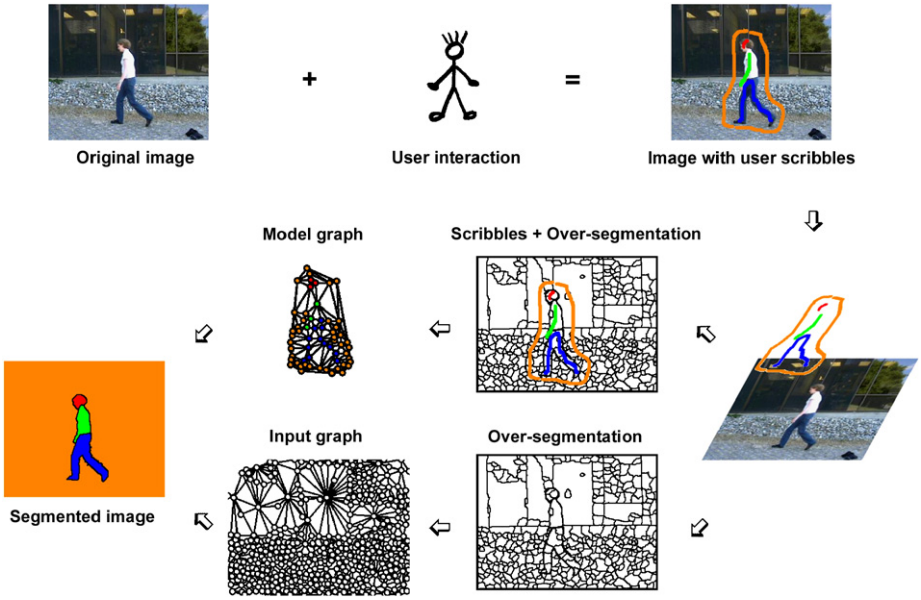


Fig. 2. Method overview. Initially, the user scribbles over the image to indicate what the objects of interest are. Next, an over-segmentation is computed from the input image. This over-segmentation originates two graphs: an input graph and a model one. In the model graph, the label of each vertex corresponds to the color of the scribble intercepting its respective region. Finally, a segmentation may be found by matching these graphs to propagate the labels from the marked regions to all non-marked ones, producing a labeling of the whole input image.

segmentation problem, involving single images, by formulating it as a matching task between (attributed relational) graphs. In this case, the input image is over-segmented to produce two attributed relational graphs (ARG), i.e., graphs in which vertices and/or edges are associated to feature vectors: the input ARG, representing all regions[1] (corresponding to the entire image), and the model ARG, representing only the regions intercepted by the scribbles (Fig. 2).

The goal is to map each input vertex to a model vertex, resulting in a many-to-one correspondence, in which the segmentation is achieved by propagating the labels given by the model vertices, as shown in Fig. 3.

To compute a match between these graphs, a cost function must be optimized in order to evaluate and choose a proper mapping among the exponential number of possible solutions. This cost function may consider appearance (e.g. average intensities/color of each region) and structural features of the image regions (e.g. relative positions among region centroids), encoded as vertex and edge attributes respectively.

In general, the evaluation of the similarity between input and model graphs can be too costly due to the presence of noise or distortions between input and model patterns, which directly affects vertex and edge attributes. Note that the interdependences represented by the graph edges turn the general graph matching problem into a challenging task, especially when we need to locate a 'small' graph within a 'large' one.

[1] Regions offer many advantages over pixel features: regions are more noise-tolerant and make constraints, such as contiguity/smoothness and adjacency, easier to formulate, as observed in [31,32].

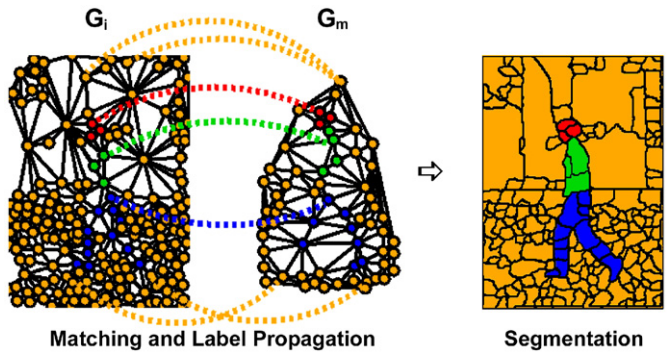


Fig. 3. Label propagation. Each scribble is characterized by a color. Each model vertex is associated to a label corresponding to the color of the scribble intercepting its respective region. During the matching step, these labels are propagated from the model G_m to the input graph G_i. This defines a complete segmentation for the image represented by G_i, by painting each pixel with the respective color assigned to the input region. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Nevertheless, when we have a 'good' initialization to align both graphs, these interdependences can be ignored and we can efficiently evaluate the spatial configurations of vertices by using deformed graphs (DG) [25], where each candidate pair of vertices (input, model) can be examined independently of the already mapped vertices. This is the key idea in the graph matching step adopted in this paper.

In our experiments, by replacing the previous tree search-based optimization algorithm [11] by the DG technique [25], we observed good matching results which were comparable to the tree search algorithm using complete model graphs, speeding up the segmentation process and allowing more interactivity with the user to produce more accurate segmentations. Moreover, we combined the DG algorithm with two post-processing steps. Firstly, by imposing a simple connectivity constraint between the segmented regions and the scribbles, we noticed that the user effort can be significantly reduced in the sense that few scribbles can produce the desired segmentation. For instance, the connectivity constraint is also present on segmentation approaches based on geodesic distance [5,26,27], random walks [12,14,17], and in the maximal similarity based region merging technique [24] by merging adjacent regions. In order to improve the boundaries of the extracted objects, the second step consisted in a pixel-based refinement, producing results which are comparable to the current state-of-the-art algorithms.

The remainder of this paper follows in Section 2 with a brief description about related works. Section 3 describes the proposed framework to match attributed relational graphs, and the optimization algorithm based on deformed graphs [25]. Section 4 presents successful results using simple scribbles, and quantitative comparison with some of the main current graph-based methods. Besides successful results for binary and multi-label segmentation for the usual case, examples illustrating the additional feature, involving multiple similar images segmentation with a single set of user seeds, are also provided. Finally, some conclusions are drawn in Section 5, besides a brief discussion about future work.

2. Related work

This section describes the main classes of methods designed for interactive segmentation relying on graphs (e.g. for applications such as photo-editing), taking into account the user intention, typically through interactions at the beginning of the segmentation process. This overview does not aim to be exhaustive, being provided to exhibit the main trends in current research.

A pioneering work involving seeds for image segmentation was due to Beucher and Meyer [6], in which the authors combined watershed with markers. Instead of using local minima to produce the regions/basins in the watershed, seeds may be provided (or found automatically) to produce the segmentation, dividing the image into the desired regions. Differently from the watersheds with markers [6], in the present work, the provided seeds are used to collect color information from the regions produced by the local minima, and structural information represented by the spatial relations, which are used to disambiguate regions having similar colors and belonging to different objects.

Currently, the main techniques using graphs for interactive segmentation are based on watershed, graph cuts, shortest paths (geodesic) and random walker.

The method based on interactive graph cuts (IGC) [7] was the first work involving graph cuts (GC) for interactive image segmentation, proposed by Boykov and Jolly. The segmentation is performed by the min-cut/max-flow algorithm. The user scribbles extract color information and are used as hard constraints. It has become very popular due to its strong mathematical foundation provided by the MAP-MRF framework. The GrabCut [28] algorithm extended the IGC method by simplifying user interaction.

A very useful segmentation benchmark, with a platform implementing important algorithms, has recently been proposed by McGuinness and Connor [23]. The authors compared important algorithms such as IGC [7], seeded region growing (SRG) [2], simple interactive object extraction (SIOX) [15] and binary partition tree (BPT) [1,29], in order to provide a good coverage of the various techniques currently available for foreground extraction, as stated in [23].

The SRG [2] method, proposed by Adams and Bischof, is very popular due to its simplicity and speed, assuming that regions of interest are characterized by connected pixels with similar colors. The SIOX [15] algorithm is also based on color information and has recently been integrated into the popular imaging program GIMP as the "Foreground Selection Tool". The BPT [1,29] algorithm is based on hierarchical region segmentation, exploiting the user interaction to split and merge regions in the tree.

Bai and Sapiro [5] proposed a method based on fast kernel density estimation [37] for the color statistics, improving the geodesic distance-based approach described in [27]. Combining the improved color statistics and the connectivity constraint imposed by the geodesic distance (between the segmented regions and the scribbles), the authors of [5] illustrate accurate segmentations using simple scribbles. Here, we also exploit the connectivity restriction in a separate post-processing step.

Grady [17] proposed random walks (RW), where each pixel is labeled based on the probability that a random walker reaches the pixel, starting from a scribble.

Variants of IGC, geodesic segmentation and RW have recently been proposed in the literature. An example is [14], where the authors formulated the interactive image segmentation problem as a statistical transductive inference, combining GC and RW. Another example is the geodesic GC [26], which uses geodesic distance information combined with edge information in a GC optimization framework.

An important work unifying GC, RW and shortest paths (geodesic) optimization techniques has recently been proposed by Couprie et al. [12], in which the authors introduced the power watershed technique, representing a new family of segmentation algorithms.

Excluding the BPT [1,29] method, all other approaches described above are based on pixels. Another important class of algorithms is represented by region merging techniques. Li et al. [21] presented a region merging method combining GC and watershed regions. Ning et al. [24] have recently proposed a novel maximal similarity based region merging (MSRM) mechanism for interactive image segmentation. The key idea of MSRM is to perform region merging between adjacent regions by exploiting an effective representation for the color statistics based on (quantized) color histograms computed from the regions. Compared to pixel-based approaches, region merging strategies aggregate robustness to noise and pixel variation in general.

Regarding the related literature, the relevance of the proposed technique has the following advantages. (1) It has been implemented as an open-source software. (2) It is trivially extensible from binary to multi-label, and from single to multiple images segmentation. (3) Good results were achieved when segmenting complex images, in which both foreground and background regions have similar colors, being comparable to the main current graph-based algorithms. (4) The graph matching step can be easily combined with the connectivity constraint between the segmented regions and the scribbles in order to reduce user effort. (5) By including a pixel-based refinement to improve the boundaries of the extracted objects, the proposed method produced accurate segmentations for a wide range of natural images.

3. Proposed framework

3.1. Segmentation by matching attributed relational graphs

Interactive segmentation relies on user hints (markers or scribbles) to segment objects in an image. In the case of binary segmentation, there are two types of scribbles, one for the foreground and the other for the background. In multi-label segmentation, the user defines as many scribbles as there are objects of interest in the image. In both cases, we consider each scribble to have a distinct color, which will also identify the referenced object during object segmentation. This prior information/labeling introduced by the user must be propagated to the non-marked regions in order to achieve a complete segmentation of the input.

In order to propagate the labels, we represent the input image data and the information encoded by the scribbles by means of attributed relational graphs (ARG) [33]. This data structure is a directed graph, in which we associate attribute vectors to vertices and edges. In the remainder of the text, all references to graphs imply a reference to an ARG.

We denote an ARG by $G=(V,E,\mu,\nu)$, in which V is a set of vertices, E is a set of (directed) edges, μ represents the vertex attributes, and ν represents the edge attributes. Cardinalities of the vertex and edge sets are denoted by $|V|$ and $|E|$ respectively.

In this paper, vertices represent image regions and edges represent spatial configurations among these regions. Also, region appearance information is encoded as vertex attributes, and the structural constraints as edge attributes. To obtain these graphs from an image, we first obtain an over-segmentation of the input image in which the contours of each object are expected to be present. In this paper, this is achieved by the watershed algorithm [34]. The resulting over-segmented image can then be used to create an input ARG and/or a model ARG, as explained next.

Input graph: We shall denote this graph by $G_i=(V_i,E_i,\mu_i,\nu_i)$ and similar subscripts will be adopted for its individual vertices and edges. It can be obtained from the input image as follows. The input vertex set V_i is defined by the watershed regions, in which there is a vertex representing each region. For the edge set E_i, there are different possibilities to represent the relations between vertices. For instance, an edge can be created to link vertices corresponding to adjacent regions. Then, vertices and edges can be attributed, as described in the next section.

Model graph: The model graph is denoted by $G_m=(V_m,E_m,\mu,\nu)$, and similar subscripts will be adopted for its individual vertices and edges. Differently from the input graph, this graph represents only specific regions of an over-segmented image: those which are intercepted by a scribble. For each of such regions, a vertex is created. If more than one scribble fall upon a single region issued from the watershed, then the dominant one (i.e. the one which intercepts more pixels from the region) can be chosen as the region label. Note that, in general, the marked regions may not induce a connected graph if edges are created based on region adjacency. By computing a Delaunay triangulation based on the considered region centroids to build the set E_m, close regions can be linked by an edge while keeping G_m connected. Finally, the attributes are computed.

3.2. ARG attributes

The choice of ARG attributes depends on the application. In our interactive segmentation experiments, we compute the same types of vertex and edge attributes for both G_i and G_m as follows:

Appearance information: The appearance information of a vertex v, denoted by $\mu(v)$, corresponds to the following attributes:

- the average intensity $\mu(v)(intensity)$ of its respective image region. Here, we focus on color images, in which we use three such values, one for each channel in the CIELAB color space, which is appropriate for computing distances between colors. This approach is significantly simpler than the color histograms used by the maximal similarity based region merging technique described in [24], each one having $16\times16\times16=4096$ bins and corresponding to a different region in the over-segmentation;
- a label $\mu(v)(label)$ identifying the class/scribble of the vertex. For vertices in G_m, this is precisely the color of the respective scribble, whereas for vertices in V_i, the label is initially undefined, being assigned later during the matching process.

Structural information: Besides the appearance information, structure is used to minimize the ambiguities caused by regions with similar intensities belonging to both foreground and background components (or distinct classes, in the case of multi-label segmentation). The structural information is represented by the spatial relations among the centroids of the regions issued from the over-segmentation. These relations are encoded by vectors in the 2D space, as shown in Fig. 4. The basic idea is that each directed edge holds a corresponding vector as its attribute. Inspired by the work described in [10], the relative positions are evaluated by the following equation, which compares two given vectors, $\vec{v_1}$ and $\vec{v_2}$, in terms of the angle between them and their lengths:

$$c_{\mathrm{vec}}(\vec{v_1},\vec{v_2})=\lambda_2\frac{|\cos\theta-1|}{2}+(1-\lambda_2)\frac{\|\vec{v_1}|-|\vec{v_2}\|}{C_S}, \tag{1}$$

where θ is the angle between $\vec{v_1}$ and $\vec{v_2}$, $|\vec{v_1}|$ and $|\vec{v_2}|$ denote the vector modulus/lengths.

The first term in Eq. (1) represents the angular cost, which assigns higher values to opposite vectors. The second term represents the modular cost, which assigns a value proportional to the difference of the vector lengths, normalized by a constant

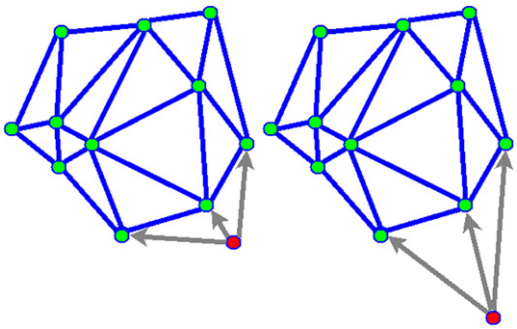


Fig. 4. For the edge attributes, we used vectors to encode spatial relations among vertices. The highlighted vertices (in red) have different coordinates, producing different vectors in terms of size and orientation. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

C_S, representing the maximum distance, to keep the computed values between 0 and 1.[2] The parameter λ_2 ranges from 0 to 1.[3]

3.3. Segmentation by matching attributed relational graphs

With these two representations at hand, the goal is to find a mapping from input to model vertices to label each input image region with a scribble color. However, by construction, the model and the input graphs present distinct topologies. Also, since it is desirable that the user inputs little effort when segmenting an image, only a few scribbles are expected over the image, which causes $|V_m|$ to be much smaller than $|V_i|$.

This mapping qualifies as an inexact homomorphism between input and model graphs (Fig. 3), a problem which has been dealt within the literature of inexact graph matching [8,10,36]. One way to solve this many-to-one mapping is by optimizing a cost function which evaluates the similarities between attributes (of vertices and edges). For instance, one can use the following general form of the cost function:

$$E = \lambda_1 \sum_{vertices} d_A + (1-\lambda_1) \sum_{edges} d_S, \tag{2}$$

where the term d_A evaluates the dissimilarities between appearance attributes of pairs of vertices (v_i, v_m), the term d_S evaluates the structural dissimilarities between pairs of edges (e_i, e_m), and the parameter λ_1 ranges between 0 and 1, weighing the influence of each term over the result.

In the general case, by considering structural information to map input to model vertices, the direct comparison between input and model edge attributes can be too costly due to a combinatorial number of all possible edge interconnections. In our method, we try to overcome this problem by using deformed graphs [25].

Matching by deformed graphs: Instead of evaluating all the interdependences represented by the edges in G_i, we propose an alternative approach which evaluates individual input vertex mappings against the model graph.

In our particular problem, because an initial correspondence is available to align both graphs (regions intercepted by the scribbles), we can use it to evaluate local deformations among attributes, without compromising the existing interdependences. This simplification is achieved by successively matching deformed graphs with the model graph, and obtaining a label for each $v_i \in V_i$ (Fig. 5). Note that, differently from the tree search technique described in [10], the classification of each input vertex v_i is done independently of the already mapped input vertices v_i', $v_i' \neq v_i$.

A deformed graph (DG) [25] represents a deformation of the model graph by a vertex issued from G_i, as if its corresponding region had fallen under a scribble during user interaction. The (local) deformation occurs in G_m with respect to the attributes of a single vertex v_m and its neighboring edge attributes, caused when simulating a change in the coordinates of v_m. Given a pair (v_i, v_m), $v_i \in V_i$, $v_m \in V_m$, the induced deformed graph, denoted by $G_d(v_i, v_m)$, is computed as follows: $G_d(v_i, v_m)$ starts as a copy of the model G_m, with the same number of vertices and edges, and the same attributes for the vertices and edges, except for the copy of model vertex v_m and its adjacent edges. The centroid coordinates and intensity attributes of the copy of v_m are replaced by that of v_i, leading to the deformed vertex v_d. Similarly, a deformed edge e_d corresponds to the edges with an endpoint at v_d, as shown in Fig. 6(c) and their attributes are recomputed to take into account the new coordinates in v_d.

By ignoring the adjacency information in E_i, the matching technique based on DG always performs comparisons among similar graphs, in the sense that both graphs have the same topology, with the same number of vertices and edges, differing only on the attributes. Moreover, only the attributes from the deformed vertex and its neighboring edges in $G_d(v_i, v_m)$ may differ from the (original) model graph G_m. Therefore, when evaluating the structural dissimilarities between $G_d(v_i, v_m)$ and the model G_m, only the deformed edges (and their corresponding model edges) have to be examined. Thus, for the evaluation of this isomorphism, Eq. (2) simplifies to

$$E(v_i, v_m) = \lambda_1 d_A + (1-\lambda_1) \sum_{deformed_edges} d_S, \tag{3}$$

where the term d_A evaluates the dissimilarities between the attributes of the deformed vertex v_d and the original model vertex v_m (Eq. (4)), the term d_S evaluates the structural dissimilarities between the deformed edges and their respective originals (Eq. (5)), and the other parameters remain the same as before. The appearance term of Eq. (3) compares vertex attributes from a pair (v_d, v_m), with v_d corresponding to an input vertex represented in G_d, as follows:

$$d_A(v_d, v_m) = \frac{\text{Euclidean distance } (\mu(v_d)(intensity), \mu(v_m)(intensity))}{C_A}, \tag{4}$$

where the Euclidean distance, between the intensity attributes of v_d and v_m, is normalized by a constant C_A, representing the maximum distance among intensity values, to keep the computed results between 0 and 1.[4] The structural term of Eq. (3) is evaluated by:

$$d_S(G_d(v_i, v_m), G_m) = \frac{1}{|E(v_d)|} \sum_{e_d \in E(v_d)} c_{\text{vec}}(v(e_d), v(e_m)), \tag{5}$$

in which $c_{\text{vec}}(.)$ is given by Eq. (1), $E(v_d)$ denotes the set of deformed edges originating from v_d, $|E(v_d)|$ denotes the cardinality, and e_m is the model edge corresponding to the deformed edge e_d. Eq. (5) is the average cost between deformed edges and their corresponding model edges.[5]

[2] In our experiments, we used $C_S = \sqrt{w^2 + h^2}$, where w is the image width and h is the image height.

[3] As in the original algorithm described in [25], we used $\lambda_2 = 0.5$ in all experiments to give the same importance to both angular and modular terms of Eq. (1).

[4] In our experiments, we used $C_A = 100$.

[5] Unless stated otherwise, we used $\lambda_1 = 0.5$ in all experiments to give the same importance to both appearance and structural terms in Eq. (3).

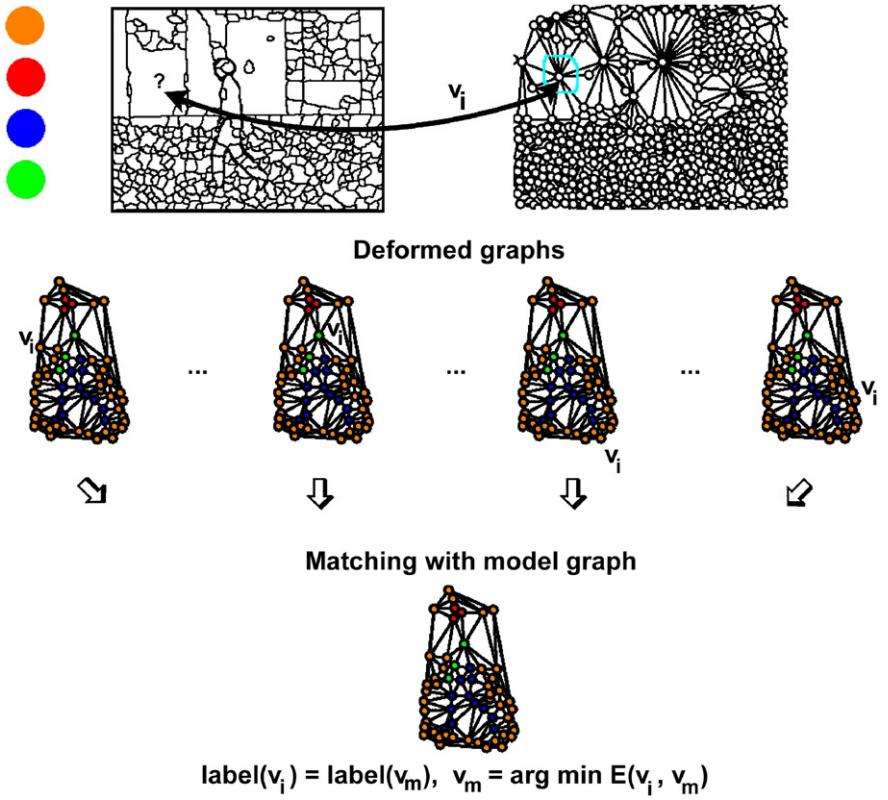


Fig. 5. Matching using deformed graphs. Because the input and model graphs can have very distinct topologies, deformed graphs are used to reduce the many-to-one mapping problem to a set of simpler inexact isomorphism problems between deformed and model graphs. The final image labeling is then given by evaluating these isomorphisms according to a greedy strategy, encoding color information and spatial configuration. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

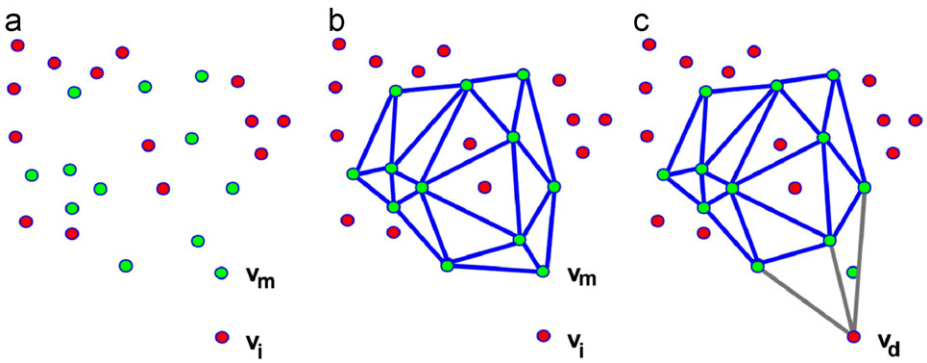


Fig. 6. (a) The model vertices (in green) are superposed to the input vertices (in red). For instance, v_m is a model vertex and v_i is an input vertex. (b) The edges denote a triangulation using the model vertices. (c) v_d is a deformed vertex, with the same coordinates as v_i, and the resulting deformed edges correspond to all (highlighted) edges with an endpoint at v_d. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3.4. Optimization

Since the aim is to map all input vertices, a set of deformed graphs must be evaluated for each $v_i \in V_i$, as shown in the pseudo-code given in algorithm *MatchingByDG*.[6]

```
MatchingByDG (G_i,G_m)
P ← ∅
for each vertex v_i ∈ V_i
        do
            c_min ← ∞;
            v_min ← NULL
            for each vertex v_m ∈ V_m
                do
                    create G_d(v_i,v_m)
                    c ← E(v_i,v_m)    ▷ compares G_d(v_i,v_m)
                    with G_m
                    if c < c_min
                        do
                            c_min ← c;
                            v_min ← v_m
            μ(v_i)(label) ← μ(v_min)(label)
            P ← P ∪ {(v_i,v_min)}
return P
```

[6] The proposed algorithm does not depend on the number of scribbles to map each input vertex to a model vertex. The same pseudo-code can be used to treat the binary (foreground/background) or the multi-label segmentation problem.

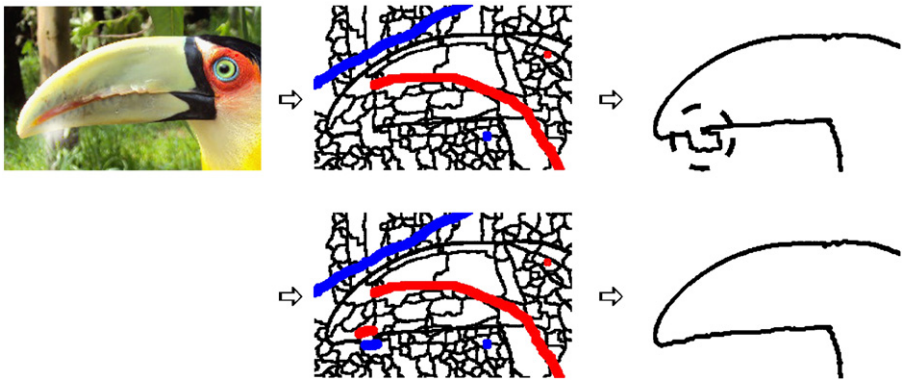


Fig. 7. Example of a coarse over-segmentation which produced an incorrect segmentation of the nose of the toucan. A correction can be achieved by manually dividing the highlighted coarse region. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Let P be a set of pairs representing a match from input to model vertices. Then the total cost of solution P is given by

$$E(P) = \sum_{(v_i, v_m) \in P} E(v_i, v_m). \tag{6}$$

In order to minimize Eq. (6), the algorithm uses a greedy strategy that relies on the individual matches between input and model vertices. The label to be assigned to each v_i should be that of the model vertex which corresponds to the DG $G_d(v_i, v_m)$ that minimizes Eq. (3), i.e., which minimizes the deformations induced in the model. Hence, we must evaluate $|V_m|$ isomorphisms between an induced DG $G_d(v_i, v_m)$ and the model graph G_m (lines 6–13).

In the end, the algorithm produces the set of pairs P and $\mu(v_i)(label) = \mu(v_m)(label)$, $\forall (v_i, v_m) \in P$, corresponding to a complete segmentation of the input image by coloring all the pixels from each input region with the respective vertex label, corresponding to a scribble color.

3.5. Computational complexity

The matching algorithm presented in the previous section has complexity in $O(|V_i||E_m|)$, which is basically determined by the number of deformed edges. Note that, for an efficient implementation, it is not necessary to rebuild $G_d(v_i, v_m)$ for each iteration. Instead, for each input vertex, the structural evaluation can be performed by traversing each (directed) model edge and each corresponding deformed edge only once. For planar model graphs, such as those generated by Delaunay triangulation, the number of traversed deformed edges is in $O(|E_m|) = O(|V_m|)$ during the classification of each input vertex, resulting in $O(|V_i||V_m|)$ edge comparisons to classify all the input vertices.

In the general tree search strategy, described in [10], an incremental solution is computed by expanding the nodes in the search tree. Each expanded node represents a pair which was included in the mapping, and its children represent all the possibilities of mapping the next input vertex. The key idea is to progressively consider more structural information, where the number of direct comparisons between input and model edges depends on the depth of the expanded node. At depth k, in order to select the cheapest model vertex for the current input vertex, $|V_m|(k-1)$ edge comparisons are performed.

The previous optimization technique [11] maintained the computational complexity in $O(|V_i||V_m|)$ by limiting the number of edge comparisons in $O(|V_m|)$ at each depth/level. During our experiments, we noticed that this modified version of the tree search [11] presented difficulties to compute the segmentations, especially when considering simple scribbles.[7] For the tested images, the original tree search technique presented a better performance when compared to the modified version [11], in which there is a total of $|V_m|(0+1+2+\cdots+|V_i|-1)$ edge comparisons, resulting in an algorithm with computational complexity in $O(|V_i|^2|V_m|)$. Therefore, in practice, the deformed graph-based algorithm decreased the computational complexity from $O(|V_i|^2|V_m|)$ to $O(|V_i||V_m|)$.

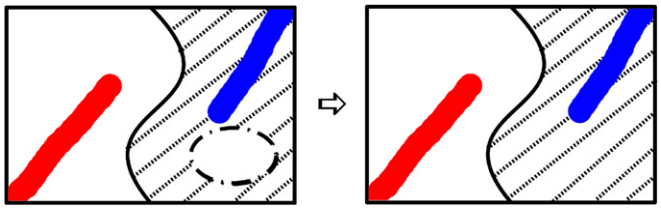


Fig. 8. (P1) Merging of regions to keep the connectivity constraint between the segmented regions and the scribbles provided by the user. In the image on the left, there is a small (solid) region, which is not connected with its corresponding (red) scribble. In the image on the right, this disconnected region is merged into the dashed region. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

3.6. Post-processing

For single images, the scribbles can be imposed as hard constraints, hence each scribbled pixel keeps its label in the final segmentation. We assume that the scribbles are correctly placed, i.e. each scribbled pixel corresponds to its correct region of interest. In our experiments, this constraint was useful to improve the results, especially for coarse over-segmentation, providing a way to manually divide a region (Fig. 7).

In order to provide ease of scribbling with minimal effort by the user and to compute accurate segmentation for object extraction, the proposed DG approach can include two post-processing steps:

(P1) Removal of spurious regions by keeping the connectivity constraint between the segmented regions and the scribbles.

[7] The examples presented in the previous work [11] required complex scribbles, which were placed near the object boundaries.

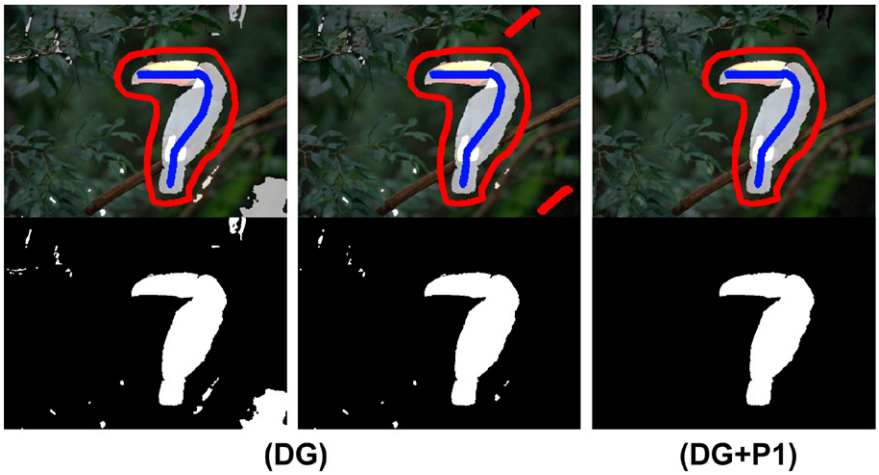


Fig. 9. Segmentation produced by the proposed DG-based approach, without and with the connectivity constraint (P1), respectively. From left to right, the two first columns represent the first case: (P1) is not imposed and the DG method requires more scribbles to remove all the spurious components from the solution (especially when considering $\lambda_1 = 0.5$).

(P2) Pixel-level refinement to improve the boundaries of the extracted objects.

In step (P1), after performing the matching algorithm (Section 3.4), the initial segmentation can be post-processed in order to guarantee that each segmented region is connected to a scribble. For instance, foreground components in the initial segmentation that are not connected to a foreground scribble are merged into the background components. Similarly, the background components that are not connected to a background scribble are merged into the foreground (Fig. 8). This simple step can significantly reduce the user effort, in the sense that less scribbles are necessary for the desired segmentation, as illustrated in Figs. 9(DG+P1) and 11(DG+P1). By considering scribbles as hard constraints, step (P1) can be used to divide coarse regions. As shown in Fig. 7, by adding scribbles, part of the (highlighted) coarse region was separated from the foreground (red) scribble to produce the correct result.

Methods combining both region merging and pixel-based approaches have been proposed in the literature (e.g. [21]), aggregating robustness (to noise and pixel variation) and accuracy (especially on the object boundaries). Following this idea, for the experiments involving binary segmentation for object extraction, we also applied the post-processing step (P2) for re-classification of pixels on a narrow band on the object boundaries, e.g. to circumvent possible imprecision in the initial over-segmentation, by performing the following heuristic. First, all pixels in a narrow band are marked as unknown pixels to indicate that they need to be re-classified (Fig. 10). In our case, the narrow band consisted of the internal and external borders of the foreground components. Then, each pixel p from the band is examined and labeled with the most similar pixel (by evaluating their colors using Eq. (4)) from all pixels at distance d from p. This step is repeated K times.[8] The improvements are illustrated in Fig. 11(DG+P1+P2).

Note that both steps, (P1) and (P2), can be executed in linear time, without affecting the computational complexity of the proposed method.

3.7. Model update

In order to achieve the desired segmentation, the user can add/remove scribbles to make corrections on the segmentation process, resulting in updates in the model graph. Fig. 12 is an example of binary segmentation, illustrating the results computed from the initial scribbles and after an additional scribble to eliminate the initial misclassified regions. After adding/removing a scribble, the model graph is rebuilt and the matching algorithm is executed using the new updated model. Then, the post-processing steps described in the previous section can be performed to improve the solution.

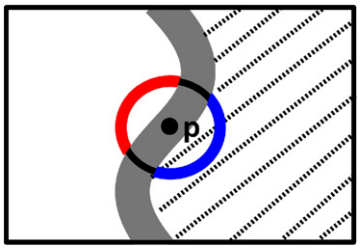


Fig. 10. (P2) Re-classification of pixels on a narrow band on the object boundaries. For each pixel p in the narrow band, p is re-classified according to the most similar pixel (in terms of color information), considering only the classified pixels that fall in a circumference with a fixed size.

4. Experiments

In order to show the benefits of the proposed approach, we tested it with different databases and real natural images, including the Berkeley Segmentation Dataset and Benchmark [22], the Microsoft GrabCut database [28], images from Ning et al. [24], Bai and Sapiro [3–5] and Levin et al. [20]. Some examples using simple scribbles are illustrated in Fig. 13.

In general, region merging approaches do not require thick scribbles to mark a large number of pixels, since regions from the initial over-segmentation can be marked by using thin scribbles. Thus, compared to pixel-based methods, more color statistics can be collected by region merging techniques when considering thin scribbles. In our implementation, we used scribbles which are 2 pixels wide to provide a clean and elegant fashion of scribbling. The results are presented with dilated scribbles for better visualization. Also, the colors of the scribbles were modified to facilitate the reading when printing in grey scale.

[8] In our case, we set $d=10$ pixels and repeated this refinement $K=10$ times.

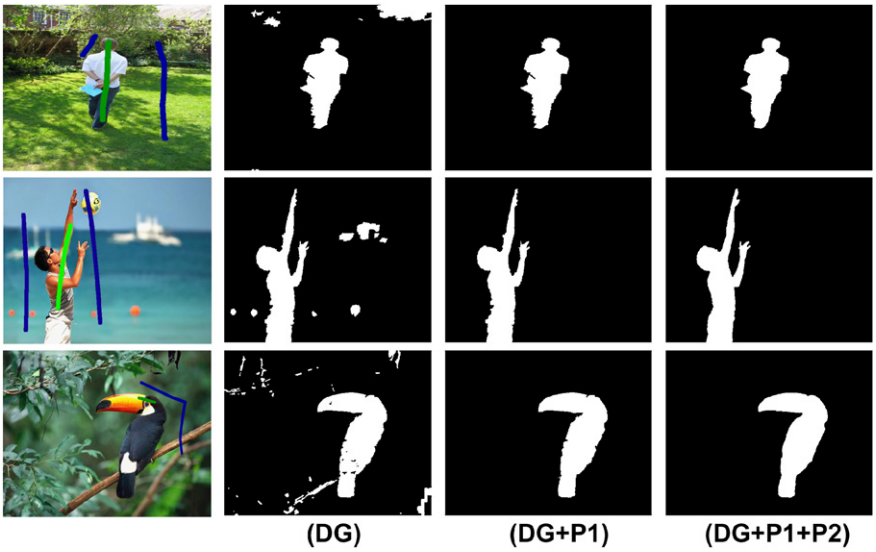


Fig. 11. Our results using images from the Grabcut database [28], Bai and Sapiro [3], and the toucan image, similar to the one used in [24], but with higher resolution (1024×768), respectively. The first column on the left shows the original images with scribbles. (DG) Computed labels by the matching algorithm described in Section 3.4. (DG+P1) Improved results after applying the post-processing step (P1) by imposing the connectivity constraint. (DG+P1+P2) Results after applying the post-processing step (P2) by improving the object boundaries.

Fig. 12. Flowers from the Berkeley database [22]. The misclassified regions (in the middle column) are corrected by adding a new scribble (in the right column). The extracted objects are highlighted.

Fig. 14 shows examples of image compositions with new backgrounds, computed by the proposed method. Fig. 15 illustrates similar segmentations using different fashions of scribbling, indicating the robustness of our method, especially when the color information is discriminative enough to separate the different components in the image.

4.1. Analysis of parameter λ_1

The parameter λ_1 weighs the influence between the appearance and the structural terms in Eq. (2). For instance, for $\lambda_1 = 0$, only structural information is used for the matching step (Section 3.4). Similarly, for $\lambda_1 = 1$, only appearance information is used to compute the segmentation. Intermediate values of λ_1 are used to combine both appearance and structural information for classification.

Additional natural images are illustrated in Fig. 16, presenting a significant variability of colors, resulting in several regions with similar colors on both foreground and background components. For these examples, the segmentation task is ambiguous, especially when using only the average colors from each region, as illustrated in Fig. 16(c). When considering only color information, these examples require more elaborated representations, such as the quantized color histograms used by the maximal similarity region merging (MSRM) technique [24]. Here, we exploit the structural information to decrease the ambiguities among similar foreground/background regions, while keeping the simplicity and the efficiency of the proposed method.

The behavior of the parameter λ_1 is illustrated in Fig. 17, in which the Jaccard index [16] was computed for different values of λ_1. As pointed out in [23], the Jaccard index is an important measure for object extraction accuracy, which can be computed by the following expression:

$$\frac{|\mathcal{G}_{\mathcal{F}} \cap \mathcal{R}_{\mathcal{F}}|}{|\mathcal{G}_{\mathcal{F}} \cup \mathcal{R}_{\mathcal{F}}|}, \tag{7}$$

where $\mathcal{G}_{\mathcal{F}}$ is the set of foreground pixels belonging to the ground-truth, while $\mathcal{R}_{\mathcal{F}}$ is the set of foreground pixels in the segmentation result. $|\mathcal{G}_{\mathcal{F}} \cap \mathcal{R}_{\mathcal{F}}|$ and $|\mathcal{G}_{\mathcal{F}} \cup \mathcal{R}_{\mathcal{F}}|$ represent the cardinalities of the intersection and the union, respectively. In Fig. 17, by testing three examples with ambiguous foreground/background regions, the best results were robust in a neighborhood around 0.5, indicating the importance of the structural information to improve the segmentations.

For a deeper evaluation, we tested the Microsoft GrabCut database [28], which is commonly used for quantitative comparisons involving different methods. This database provides the labeling-Lasso, where the unknown pixels corresponds to a roughly symmetric narrow band along the boundaries of the foreground object, as shown in Fig. 18 (middle column). For this particular type of seeds, algorithms such as those based on adaptive thresholding [14,18] can exploit the same effect of the skeleton of the unknown regions to improve the segmentations. In order to avoid any bias, instead of using the labeling-Lasso, Couprie et al. [12] proposed the use of asymmetrically eroded seeds, such as the one shown in Fig. 18 (right column).

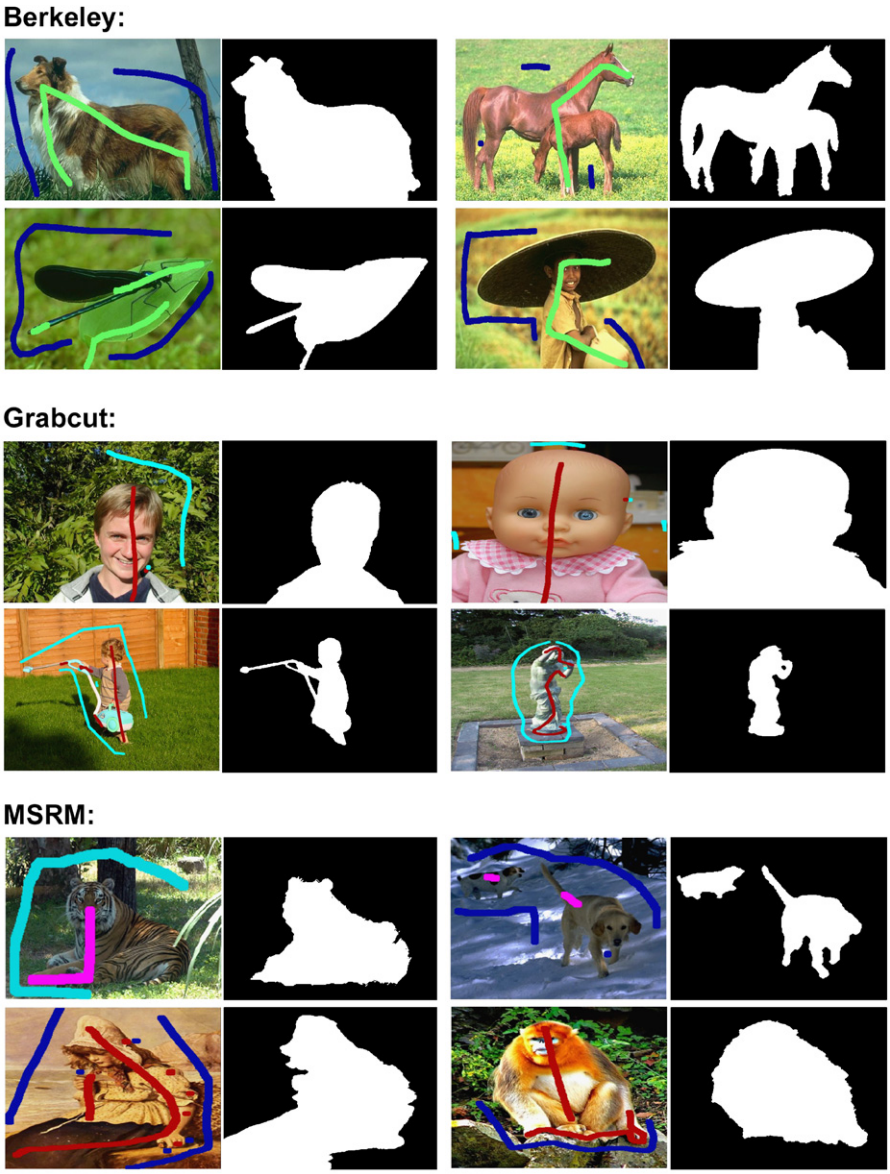


Fig. 13. Our results using images from Berkeley Image Segmentation and Benchmark [22], GrabCut [28], and MSRM database [24]. For each example, the image on the left shows the original image with scribbles, and the image on the right shows the computed labels by DG+P1+P2.

Fig. 14. Examples of background replacement using images from different databases: Berkeley [22], GrabCut [28], and Bai and Sapiro [3,5], respectively. Left column: original images with scribbles. Middle column: labels computed by the proposed approach (DG+P1+P2). Right column: compositions using new backgrounds.

Fig. 15. Flowers from the Berkeley database [22] and three different ways of placing the scribbles. The respective segmented regions (outlined in white) were computed by DG+P1+P2.

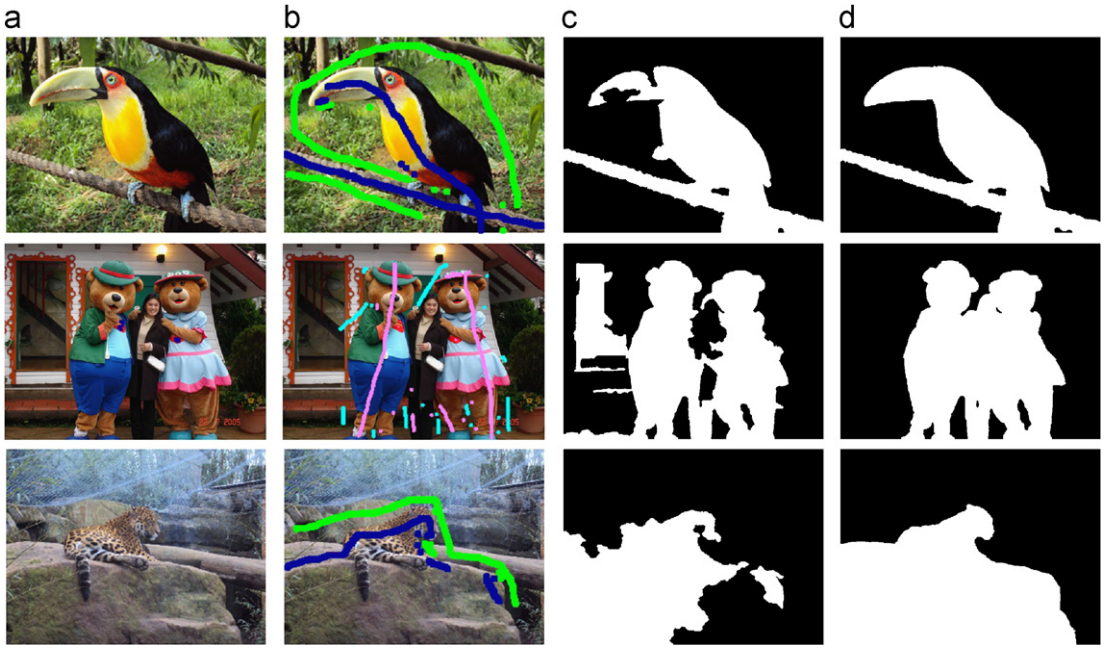


Fig. 16. Additional segmentation examples. (a) Input images: bird, bears and jaguar. (b) Scribbles over the input images. (c) Computed labels using only the appearance information ($\lambda_1 = 1.0$). (d) Computed labels using both the appearance and the structural information ($\lambda_1 = 0.5$). The above results illustrate the importance of the structural component in Eq. (2). The segmentations were computed by DG+P1+P2.

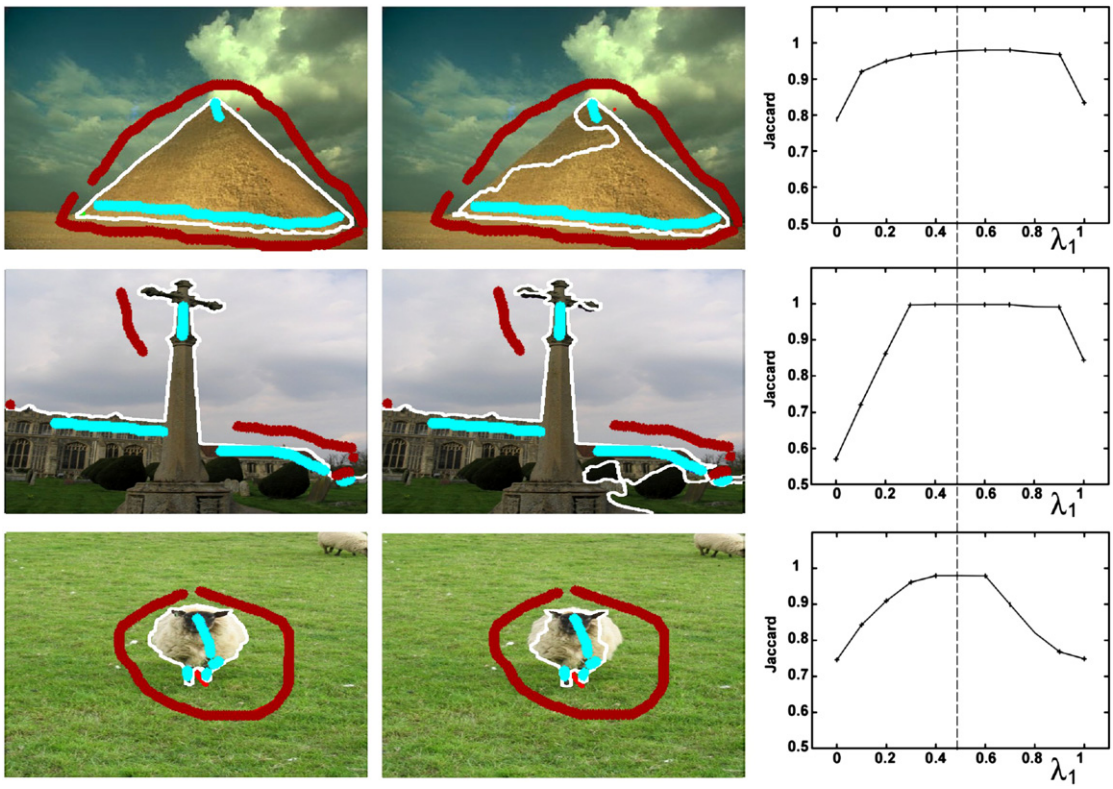


Fig. 17. Pyramid from the Berkeley database [22]; cross and sheep from the Grabcut database [28]. For each example, we tested the same scribbles for different values of λ_1, by using DG+P1+P2. The segmentations are outlined in white. (a) Segmentations by using both the appearance and the structural information ($\lambda_1 = 0.5$). (b) Using only the appearance information ($\lambda_1 = 1.0$). (c) Jaccard indices for different values of λ_1.

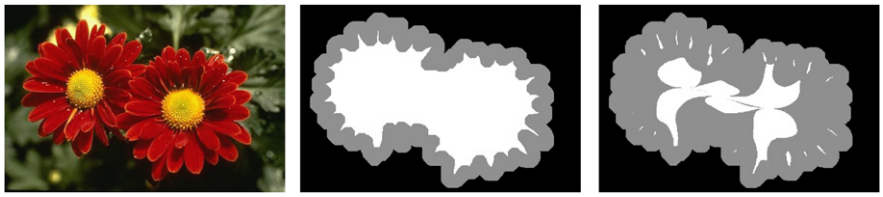


Fig. 18. Left column: flowers from the Berkeley database [22]. Middle column: respective labeling-Lasso provided by the GrabCut database [28]; the unknown pixels correspond to a symmetric narrow band (in grey) along the boundaries of the flowers. Right column: the corresponding asymmetrically eroded seeds from Couprie et al. [12].

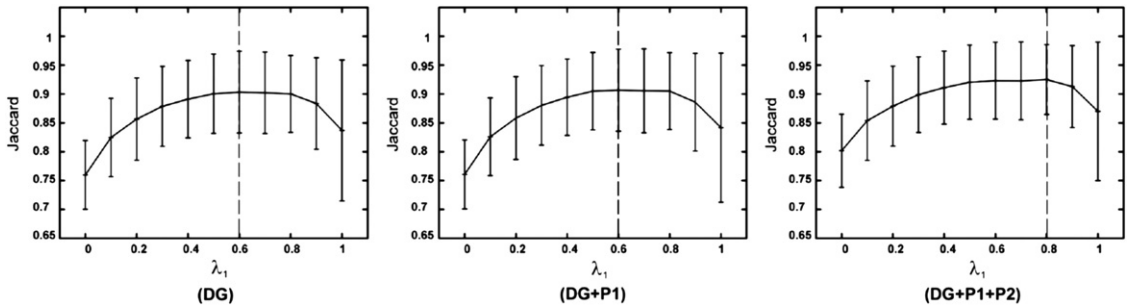


Fig. 19. The average Jaccard indices and the respective standard deviations for the fifty images from the GrabCut database [28], using the eroded seeds provided by Couprie et al. [12].

Fig. 19 shows the average Jaccard indices, for $\lambda_1 = 0.0, 0.1, \ldots, 1.0$, using the asymmetrically eroded seeds provided by Couprie et al. [12], and considering the three cases: (DG) the matching algorithm (Section 3.4) without any post-processing step; (DG+P1) combined only with post-processing step (P1); and (DG+P1+P2) combined with both (P1) and (P2). The largest averages were: (DG) 0.9033 for $\lambda_1 = 0.6$; (DG+P1) 0.9066 for $\lambda_1 = 0.6$; and (DG+P1+P2) 0.9250 for $\lambda_1 = 0.8$. Nevertheless, all three cases resulted in good average object accuracy when using $\lambda_1 = 0.5$ (0.9005, 0.9049 and 0.9204, respectively). Thus, for the comparison with the current state-of-the-art techniques, we test our approach with the fixed parameter $\lambda_1 = 0.5$.

4.2. Multi-label segmentation and image matting

The proposed formulation can also be applied to the multi-label segmentation problem in a straightforward manner. As shown in Fig. 20, the proposed approach has a good performance, even in the presence of multiple objects with similar colors, since the initial over-segmentation included the important boundaries in these examples.

In order to extract objects with complex boundaries, such as hair strands and animal fur, image matting [20,35] is suitable to reconstruct the foreground/background components and the alpha value (transparency) of each pixel. In the image matting problem, the interaction can be performed by using trimaps which divide the image into three regions: foreground, background and unknown regions. Our approach can also be used to automatically generate such trimaps, in which a narrow band is automatically spanned across the current foreground/background boundaries to define the unknown regions. Following the same idea of the fixed width band described in [5], Fig. 21 shows examples in which our approach can minimize the effort by avoiding the need to track all the object boundaries when building the complete trimaps manually.

As observed in [20], for complex cases, especially when the objects have too many holes, image matting can also be performed by using scribbles, which can require a significant effort, as suggested by the examples presented in [20]. For the cases in which the trimaps are preferable, such as Fig. 22, our approach can simplify the scribbles used for image matting.

In all matting experiments, we used the Robust Matting [35] binaries provided by the authors, which produced accurate matting results.

4.3. Comparison with current state-of-the-art

For a quantitative comparison with other methods, we used the 50 images from the Microsoft GrabCut database [28]. We considered different techniques, including pixel-based and region-based methods, for a good coverage of the literature. More specifically, we used the source codes provided by the authors of the power watershed (PW) [12], the maximal similarity based region merging technique (MSRM) [24], and random walker (RW) [17]. For the interactive graph cuts (IGC) [7], seeded region growing (SRG) [2], simple interactive object extraction (SIOX) [15] and binary partition tree (BPT) [1,29], we used the segmentation tool due to McGuinness and Connor [23], combined with a context creator utility provided by the authors of [23] to load the seeds from pre-existing image files. All these implementations are freely available, corresponding to recent important works [12,17,23,24].

In order to better exploit the capabilities of the well-known IGC method [7], we tested it with fixed parameter $\sigma = 3.5$,[9] and with the best σ for each individual image (considering $\sigma = 0.5, 1.0, 3.5, 7.0, 10.0, 15.0$ in the segmentation tool provided by McGuinness and Connor [23]), corresponding to the largest object accuracy (Jaccard index). The parameter σ is used in the boundary penalty function described in the IGC paper [7], corresponding to the distribution of noise among neighboring pixels. For the remaining competing approaches, we did not set any parameter.

Table 1 shows the results for the symmetrically and asymmetrically eroded seeds: the Lasso form [28] and the ones provided

[9] When considering IGC [7] with fixed parameter in the segmentation tool provided by [23], $\sigma = 3.5$ produced the largest average Jaccard indices in our experiments with the GrabCut database [28].

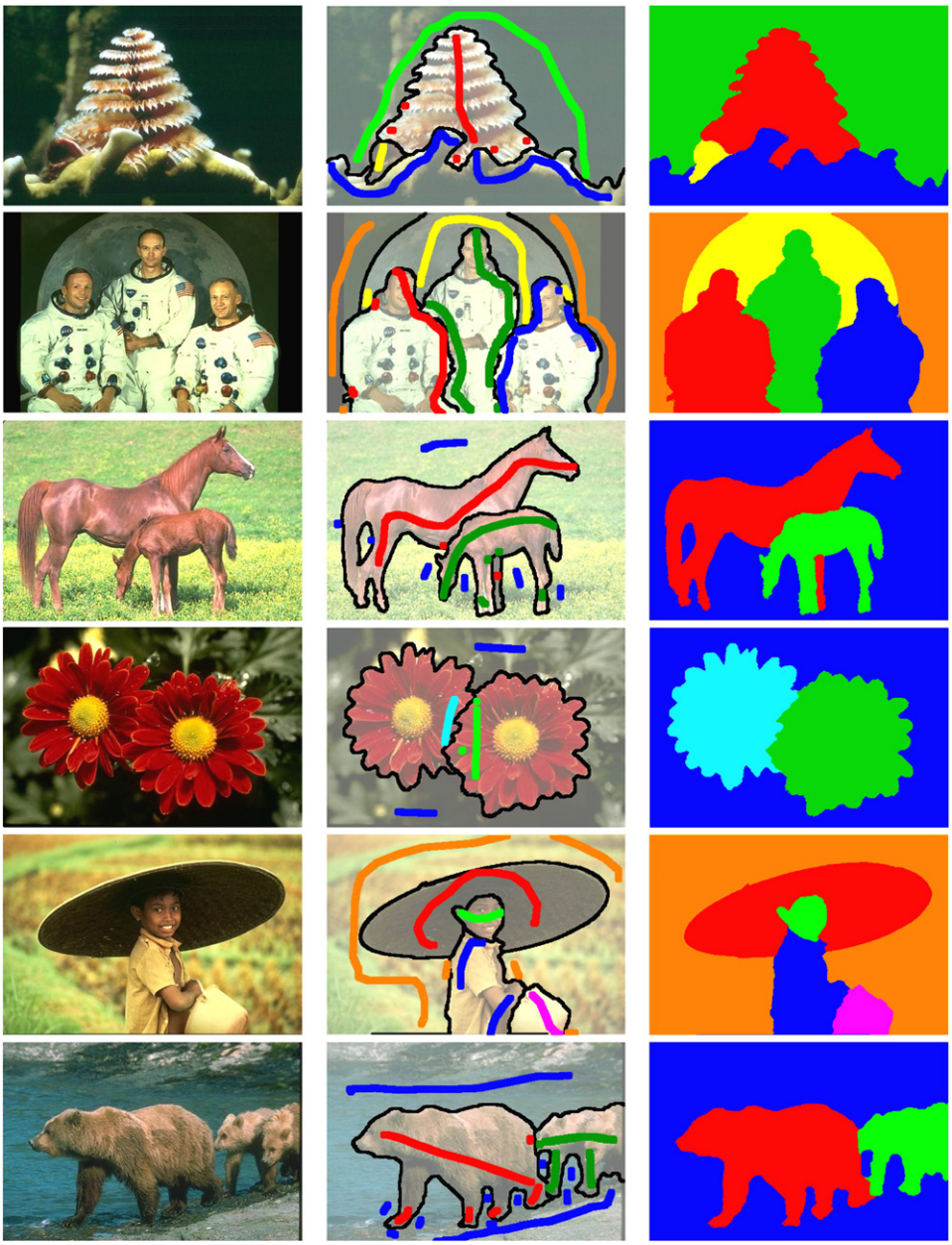


Fig. 20. Examples from the Berkeley database [22] for multi-label segmentation, including multiple objects with similar colors. The results were computed by DG+P1, using only the connectivity constraint (P1). Left column: original images. Middle column: scribbles and segmented regions outlined in black. Right column: computed labels. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

by Couprie et al. [12], respectively. When compared to the Lasso form, the asymmetrically eroded seeds from Couprie et al. [12] correspond to smaller markers for the foreground, as shown previously in Fig. 18.

The quantitative results produced by the tree search algorithm [10], considering complete (TSC) and triangulated (TST) model graphs, were also included in Table 1. For both TSC and TST, we considered a fixed parameter α (corresponding to the largest average Jaccard index) and the best α (corresponding to the largest Jaccard index for each image) for Eq. (1) of the previous work described in [11].[10]

In Table 1, our approach DG+P1+P2 with $\lambda_1 = 0.5$ (Eq. (2)) produced the largest average Jaccard index among all the considered methods for both types of seeds. Moreover, for the smallest set of seeds [12], the performance of DG (using triangulated model graphs, without any post-processing step) was comparable to TSC (using complete model graphs), but within a lower complexity time, indicating the robustness of the proposed method according to seed quantity. Differently from the DG technique, the TST method presented a poor performance (comparable to SRG), indicating insufficient structural information by the triangulated model graphs when considering the tree search technique [10] and 'small' seeds. For TST, the largest average Jaccard index was achieved when using $\alpha = 0.1$, indicating that increasing the influence of the structural term was not enough to produce good results. Moreover, by choosing the best α, the

[10] The parameter α in Eq. (1) of [11] has the same role as our parameter λ_1, where lower values for α correspond to larger weights to the structural term.

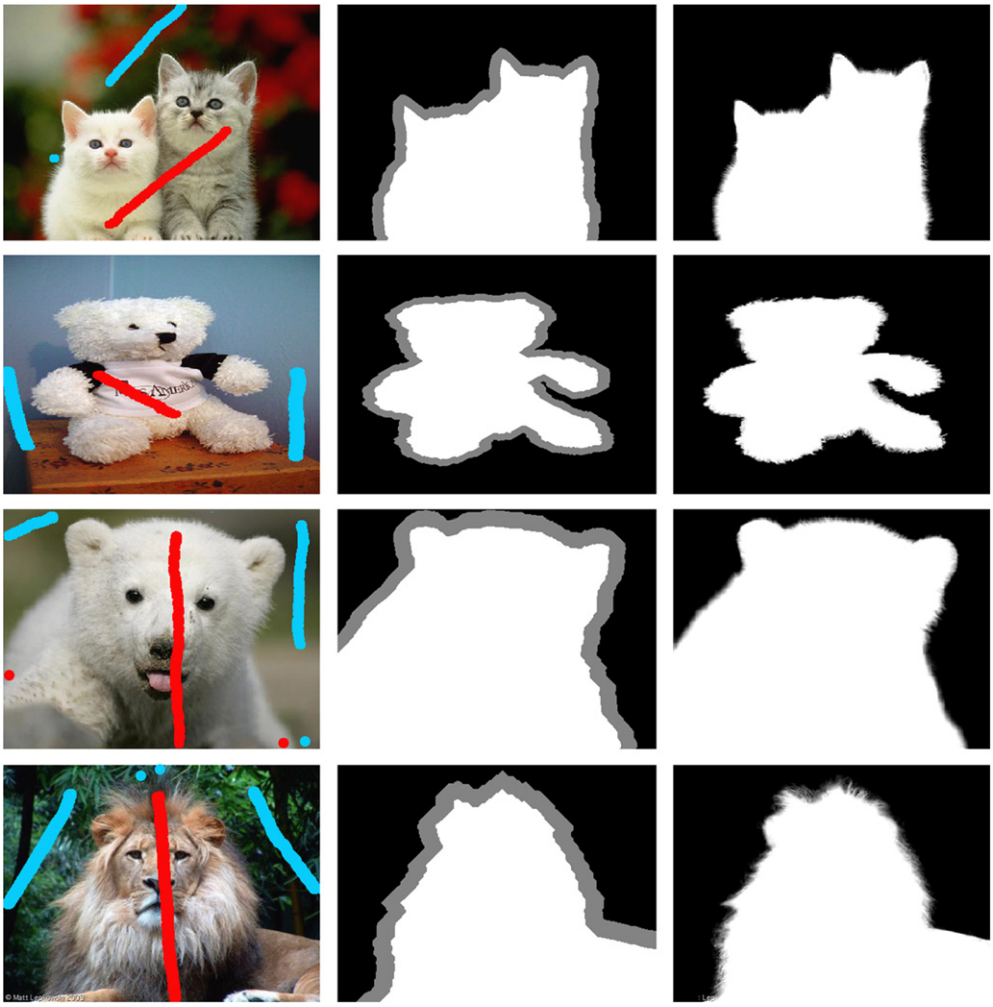


Fig. 21. Input images from Bai and Sapiro [4,5]. Left column: original images with scribbles. Middle column: automatically computed trimaps by DG+P1+P2. Right column: alpha mattes computed by Robust Matting [35] using the trimaps from the middle column.

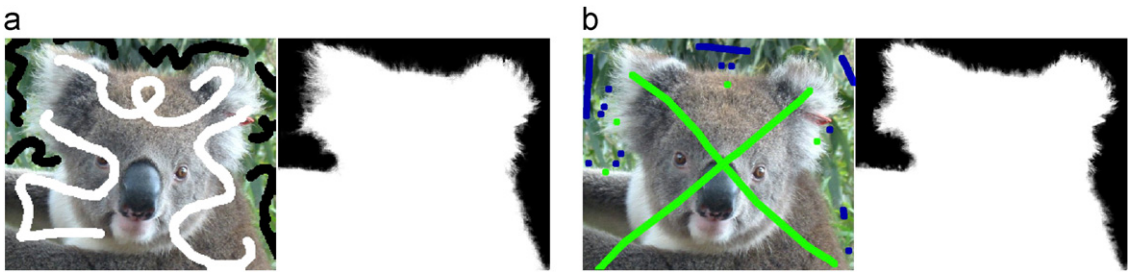


Fig. 22. (a) Original image with scribbles and the corresponding alpha matte, both images were extracted from the paper due to Levin et al. [20]. (b) Original image with scribbles and the respective alpha matte computed by the Robust Matting [35] using the automatically computed trimap by DG+P1+P2.

stability of TST was not improved, which presented large standard deviations for the object accuracy.

In the ground-truth provided by the Grabcut database [28], grey level 255 corresponds to the foreground objects, 0 to the background components, and 128 to the mixed foreground-background pixels. In all experiments, we discarded the mixed pixels from the (binary) Jaccard index computations for the object accuracy analysis, since this binary index considers only full foreground/background pixels. Moreover, based on the binary Jaccard index used for object accuracy, the authors of [23] proposed the fuzzy boundary accuracy to evaluate the segmentation results. One way to use it on the Grabcut database is to interpret the mixed/unknown pixels in the ground-truth as part of the boundary, which may increase the degree of 'uncertainty' in the boundary. Following this idea, we also present the respective fuzzy boundary accuracies in Table 1, in which we considered (besides the internal boundaries of the ground truth) the mixed regions for the corresponding fuzzification.[11] Note that, by considering this boundary accuracy

[11] As described in [23], we used $\sigma = 4$ for the fuzzification.

Table 1
Quantitative comparison using the Microsoft GrabCut database [28], with the symmetrically (the Lasso form [28]) and the asymmetrically eroded seeds (provided by Couprie et al. [12]). For the object and boundary accuracy measures, we used the Jaccard index [16] and the fuzzy boundary accuracy [23], respectively.

Method	Object accuracy (Jaccard)		(Fuzzy) Boundary accuracy	
	Average	Standard deviation	Average	Standard deviation
Symmetric seeds (Lasso)				
SIOX [15]	0.8852	0.1718	0.6338	0.1935
SRG [2]	0.8881	0.0860	0.5409	0.1797
MSRM [24]	0.9017	0.0856	0.6082	0.1467
IGC [7], $\sigma=3.5$	0.9275	0.0651	0.6376	0.1454
RW [17]	0.9324	0.0659	0.6547	0.1432
PW [12]	0.9341	0.0636	0.6653	0.1384
IGC [7],best σ	0.9366	0.0621	0.6661	0.1452
BPT [1,29]	0.9533	0.0531	**0.7266**	0.1230
DG, $\lambda_1=0.5$	0.9102	0.0835	0.6366	0.1264
DG+P1, $\lambda_1=0.5$	0.9429	0.0508	0.6898	0.1075
DG+P1+P2, $\lambda_1=0.5$	**0.9536**	0.0501	0.7239	0.1137
TSC, $\alpha=0.5$	0.9444	0.0471	0.6999	0.1012
TSC, best α	0.9525	0.0430	0.7305	0.0781
TST, $\alpha=0.1$	0.9415	0.0440	0.6888	0.0949
TST, best α	0.9438	0.0429	0.6975	0.0966
Asymmetric seeds (Couprie et al. [12])				
SRG [2]	0.8378	0.1272	0.4948	0.2010
RW [17]	0.8752	0.0986	0.5471	0.1936
IGC [7], $\sigma=3.5$	0.8784	0.0946	0.5613	0.1834
SIOX [15]	0.8830	0.1232	0.5863	0.1956
PW [12]	0.8891	0.0881	0.5842	0.1705
IGC [7], best σ	0.8955	0.0889	0.5948	0.1831
MSRM [24]	0.9017	0.0856	0.6082	0.1467
BPT [1,29]	0.9188	0.0688	**0.6368**	0.1717
DG, $\lambda_1=0.5$	0.9005	0.0687	0.5747	0.1422
DG+P1, $\lambda_1=0.5$	0.9049	0.0672	0.5846	0.1371
DG+P1+P2, $\lambda_1=0.5$	**0.9204**	0.0644	0.6255	0.1494
TSC, $\alpha=0.5$	0.9080	0.0635	0.5927	0.1274
TSC, best α	0.9195	0.0593	0.6142	0.1318
TST, $\alpha=0.1$	0.8485	0.1040	0.4879	0.1516
TST, best α	0.8546	0.1056	0.4985	0.1526

measure, the BPT technique [1,29] presented the highest fuzzy boundary accuracy. Table 2 presents additional measures, which are the same indices used in [12], in which our method presented the best performance for the smallest set of seeds from Couprie et al. [12].

For the tree search algorithms (TST and TSC), the input edges were created between adjacent regions in the over-segmentation. For all the experiments involving the MSRM approach, we used watershed for the initial segmentation for a fair comparison, which significantly increased its computational time (see Table 5 in next section).

The region-merging techniques, MSRM [24], BPT [1,29] and DG+P1+P2 with $\lambda_1=0.5$, produced the largest average accuracy indices for the eroded seeds from Couprie et al. [12]. By using 'small' sets of seeds, besides the robustness to noise and color variability, a larger number of pixels can be marked by the region merging methods, when compared to the pixel-based approaches, providing more information for the classification.

Fig. 23 illustrates a qualitative comparison, in which SRG and SIOX failed for all the three additional examples. Although RW produced a good result for the jaguar example, BPT, TSC+P1 and DG+P1 provided more consistent results for the other two examples with the given scribbles, by producing less leaking effects and less missing regions. Excluding SRG and SIOX, the

Table 2
Quantitative comparison using the Microsoft GrabCut database [28], testing both symmetrically and asymmetrically eroded seeds, with the same accuracy indices used in [12]: Rand Index (RI), Global Consistency Error (GCE), Variation of Information (VoI), and Boundary Error (BE). Good segmentations correspond to high RI, low GCE, low VoI and low BE.

	RI	GCE	VoI	BE
Symmetric seeds (Lasso)				
SIOX [15]	0.9451	0.0378	0.2928	5.1512
SRG [2]	0.9523	0.0404	0.3022	4.0483
IGC [7], $\sigma=3.5$	0.9680	0.0265	0.2235	3.3025
RW [17]	0.9704	0.0241	0.2090	3.1617
PW [12]	0.9706	0.0247	0.2102	2.8888
MSRM [24]	0.9719	0.0235	0.2099	2.4678
BPT [1,29]	0.9781	0.0176	0.1695	**1.7780**
DG+P1+P2, $\lambda_1=0.5$	**0.9784**	**0.0175**	**0.1692**	1.9124
Asymmetric seeds (Couprie et al. [12])				
SRG [2]	0.9342	0.0527	0.3692	5.9340
RW [17]	0.9457	0.0430	0.3113	5.8707
IGC [7], $\sigma=3.5$	0.9483	0.0420	0.3059	5.3369
PW [12]	0.9518	0.0400	0.2930	4.8480
SIOX [15]	0.9543	0.0369	0.2852	4.7063
MSRM [24]	0.9594	0.0338	0.2666	3.6166
BPT [1,29]	0.9634	0.0299	0.2395	3.2795
DG+P1+P2, $\lambda_1=0.5$	**0.9653**	**0.0282**	**0.2331**	**3.1799**

remaining competing methods basically required more scribbles to achieve good results, such as those shown in Fig. 16(d), which were produced by DG+P1+P2 with $\lambda_1=0.5$.

4.4. Running time

The experiments were carried out in a computer with an Intel Core i3 2.13 GHz processor and 4 GB of RAM. Table 3 illustrates the running times for object extraction. For each entry, we present the image dimensions (width and height), the number of input and model vertices ($|V_i|$ and $|V_m|$, respectively), and the computational times for: building the input and model graphs (G_i and G_m, respectively), the graph matching algorithm (Section 3.4) and the post-processing steps (Section 3.6). Note that the largest computational times are required for building G_i, mainly due to a pre-processing step which was performed to reduce the number of watershed regions, by merging small[12] regions with its most similar adjacent region (in terms of appearance/color). This may slow down the launching of the interactive segmentation program, but it was very important to reduce the computational times for the proposed approach, which considerably reduced the sizes of the input and model graphs, as shown in Table 4, without affecting the interactivity with the user during the segmentation process. In Table 3, although our code was implemented in Java, which is often slower than a C++ implementation, note that the model graph update and the graph matching algorithm presented fast performances, justifying the fact that our method can be used for the interactive image segmentation problem.

Table 5 compares the running times of different approaches. SRG [2], IGC [7] and PW [12] were the fastest implementations (in C++), while the tree search methods [11] TST and TSC were the slowest ones. Note that only the BPT running time included the pre-processing time, required to compute the binary trees. In our experiments, the user interaction of BPT was very fast, being comparable to the other C++ implementations like IGC [7] and

[12] In our implementation, we merged small regions with area <25 pixels.

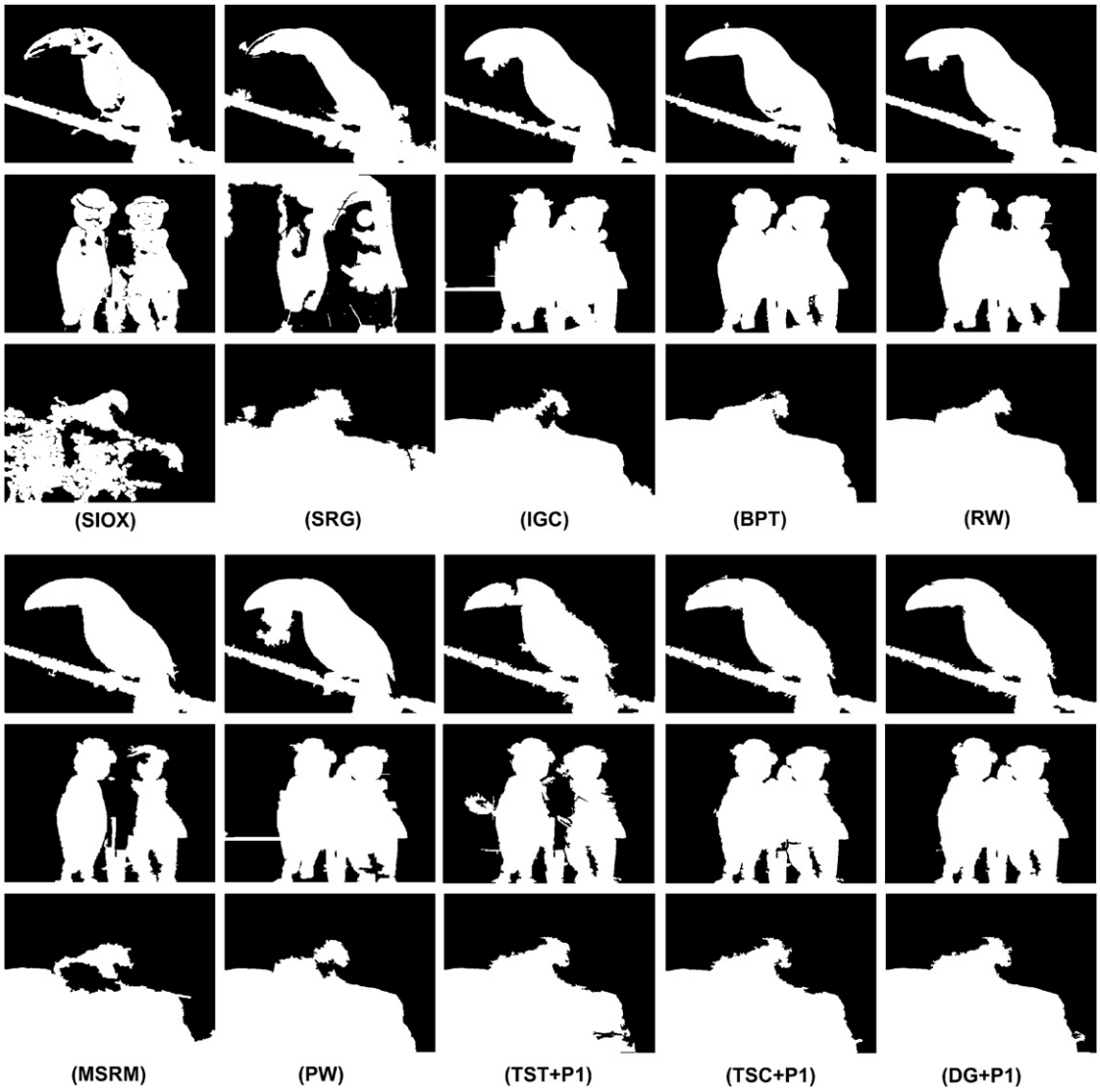


Fig. 23. Qualitative comparison using the additional images and the respective scribbles from Fig. 16. The matching techniques TST, TSC and DG were combined only with the post-processing step (P1).

PW [12]. DG, TS [11] and MSRM [24] did not include the over-segmentation computation time.

4.5. Segmentation of multiple similar images

The proposed approach can also segment multiple similar images. In this case, an important question arises: "Would it be possible to take advantage of the same user scribbles applied to one specific image in order to segment another similar image?".[13] Specifically, given two similar images A and B (e.g. corresponding to different frames from a video sequence), and seeds marking the regions of interest in A, the goal is to segment B by using the same prior information represented by the scribbles drawn over A. In general, direct use of scribbles from A on B does not result in good segmentation due to possible deformations between A and B, especially when dealing with articulated objects (Fig. 24). Thus, a flexible representation for the scribbles is necessary in order to produce a successful segmentation for B.

By using attributed relational graphs as described in Section 3, the proposed approach represents scribbles from A and image B as a model graph and an input graph, respectively (Fig. 25). Then, for instance, by performing the algorithm *MatchingByDG* (Section 3.4) to obtain a solution for the graph matching problem, encouraging segmentation results were achieved for roughly aligned graphs.

In Fig. 26, we considered five different frames (63–67) from the video sequence of a person walking straight from [30], assuming frame 65 as the model graph to segment all five given frames. In this example, the results from our approach outperformed the ones achieved by directly applying the scribbles with usual methods to segment each frame. For the third row in Fig. 26, we tested DG+P1+P2, $\lambda_1 = 0.5$ and PW [12] for the usual single image segmentation by directly applying the scribbles on each frame. Both methods produced similar leaking effects, especially on the legs.

Fig. 27 illustrates our results on the Tsukuba images, which are very popular for stereo benchmarking. Differently from most

[13] Note that, differently from [13], which relied on manual segmentations, our goal is to take advantage of the scribbles to propagate the segmentation to other similar images.

Table 3
Running times for initial segmentation computed by graph matching and for post-processing using Figs. 13–16, 20–22.

Image	Size	$\lvert V_i \rvert$	$\lvert V_m \rvert$	Computational times (in ms)				
				G_i	G_m	Matching	P1+P2	Total
Binary segmentation								
berk-anemone	480 × 320	1752	170	3588	78	1863	2599	8128
berk-flowers	480 × 320	1780	30	3341	42	365	2604	6352
berk-dog	480 × 320	1431	185	3768	72	1638	2071	7549
berk-horses	480 × 320	1015	63	3835	46	485	3503	7869
berk-insect	480 × 320	2144	244	3541	87	3135	2463	9226
berk-boy	480 × 320	1976	186	3599	77	2200	2175	8051
grab-person1	600 × 450	1963	181	6585	80	2208	2851	11724
grab-person2	600 × 450	2350	117	6355	139	1663	3702	11859
grab-doll	462 × 549	2836	128	5612	129	2175	3234	11150
grab-child	1024 × 768	6214	314	19 340	252	11 032	8224	38 848
grab-statue	768 × 1024	5175	315	20 322	236	9892	13 412	43 862
msrm-tiger	264 × 192	314	72	1203	69	211	1709	3192
msrm-dogs	335 × 295	1135	102	2092	79	757	1936	4864
msrm-girl	303 × 397	982	196	2922	107	1208	2053	6290
msrm-monkey	360 × 414	1136	145	3317	105	1013	3724	8159
add-bird	640 × 480	2659	389	7471	142	5956	6526	20 095
add-bears	800 × 600	5557	386	10 810	186	11 823	9838	32 657
add-jaguar	640 × 480	2019	182	7583	83	2300	7394	17 360
bai-boy	255 × 308	743	78	1840	31	421	2075	4367
bai-cat3	535 × 412	2465	67	4384	156	967	3026	8533
bai-toy	338 × 450	1344	73	3650	62	671	2325	6708
bai-bear	450 × 344	1863	107	3573	47	1263	2278	7161
bai-lion	400 × 300	1013	92	2636	47	593	2090	5366
levin-teddy	486 × 416	1716	191	4527	109	2028	4508	11 172
Multi-label segmentation								
berk-anemone	480 × 320	1752	249	3597	193	2606	2988	9384
berk-astronauts	480 × 320	1256	269	3410	196	1988	3607	9201
berk-horses	480 × 320	1015	106	3835	161	697	3682	8375
berk-flowers	480 × 320	1780	63	3355	148	581	4113	8197
berk-boy	480 × 320	1976	241	3589	193	2755	3222	9759
berk-bears	480 × 320	1109	161	3804	171	1116	3306	8397

Table 4
Number of original watershed regions and the corresponding reduced amount after the pre-processing step.

Image	Number of watershed regions	
	Original	Pre-proc
berk-dog	14,420	1431
berk-horse	17,471	1015
berk-insect	9620	2144
berk-boy	11,205	1976
berk-flowers	11,324	1780
grab-person1	27,796	1963
grab-person2	27,705	2350
grab-doll	15,446	2836
grab-child	84,025	6214
grab-statue	90,294	5175
msrm-tiger	6052	314
msrm-dogs	7603	1135
msrm-girl	12,296	982
msrm-monkey	16,258	1136
add-bird	29,117	2659
add-bears	34,591	5557
add-jaguar	34,603	2019
bai-cat3	14,580	2465
bai-toy	14,767	1344
bai-bear	9938	1863
bai-lion	10,993	1013
levin-teddy	20,396	1716

Table 5
Average running times by using the grabcut database [28] with the asymmetric seeds by Couprie et al. [12].

Method	Mean time (s)
IGC [7], $\sigma = 3.5$	0.4259
SRG [2]	0.4593
PW [12]	1.2759
SIOX [15]	1.4072
RW [17]	1.8866
BPT [1,29]	2.9126
DG, $\lambda_1 = 0.5$	11.4762
DG+P1, $\lambda_1 = 0.5$	12.7314
DG+P1+P2, $\lambda_1 = 0.5$	16.5837
MSRM [24]	153.7778
TST [11], $\alpha = 0.1$	187.2444
TSC [11], $\alpha = 0.5$	386.1613

of the usual interactive segmentation approaches, by using a model graph, we produced encouraging results by avoiding most of the segmentation errors due to incorrect placement of the seeds. More challenging results are shown in Figs. 28 and 29, illustrating large object displacements, camera position and color variability. Despite the differences among the images, the model created for the first one is successfully used to segment the others.

5. Conclusions

In this paper, we have proposed an interactive algorithm for image segmentation, which produced accurate results for a wide range of natural images. The core of the technique corresponds to

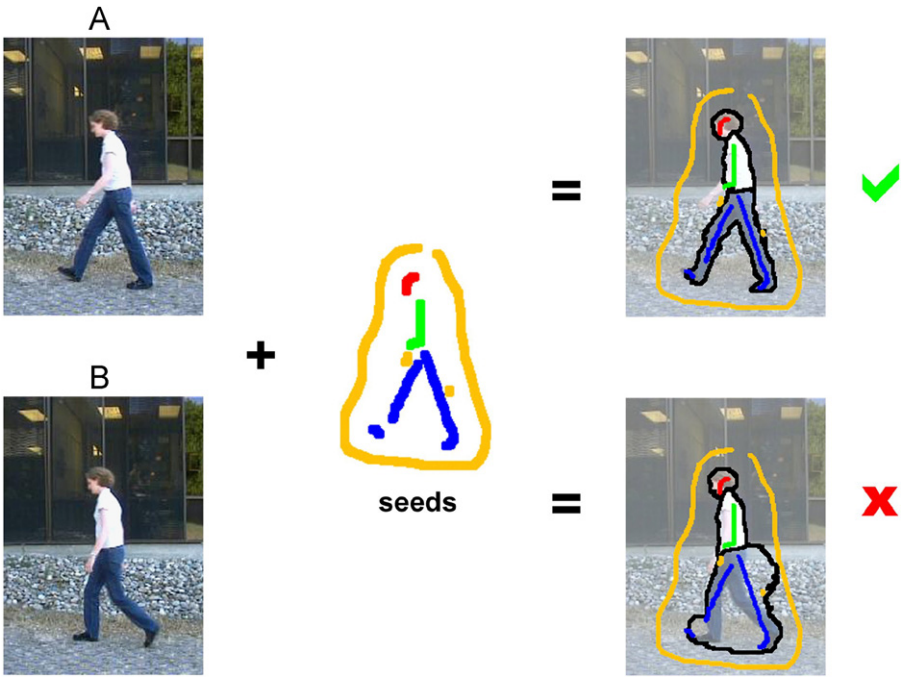


Fig. 24. Motivation: "Could we use the scribbles drawn over *A* to segment a similar image *B*?". In general, given two similar images *A* and *B* and scribbles marking the regions of interest in *A*, their direct application to the segmentation of *B* does not produce good results. This is because markers may fall over wrong regions in *B*. Results shown were both produced by our DG+P1+P2 algorithm for segmenting single images.

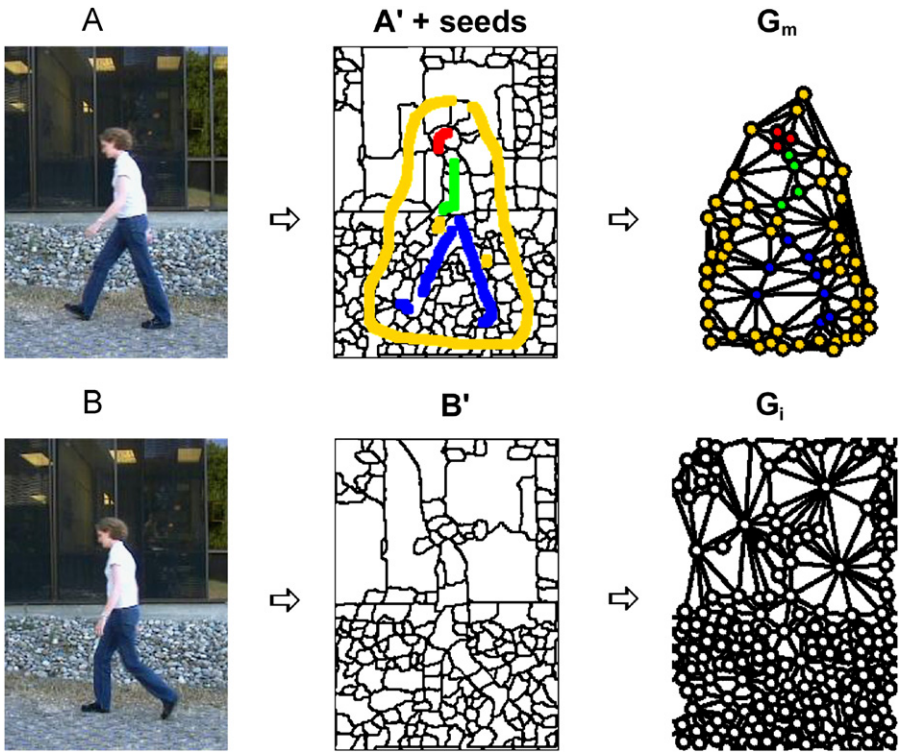


Fig. 25. Model and input graphs representing similar images, *A* and *B*, respectively. The corresponding over-segmentations are A' and B'. The scribbles are placed over A', indicating the regions of interest, resulting in the model graph G_m. The input graph G_i represents all the regions in B'.

a framework to match attributed relational graphs by exploiting the spatial relations among vertices.

For the optimization, we exploited deformed graphs [25], which can be applied to multi-label segmentation (representing multiple objects or multiple object parts) without any change in the core algorithm, in contrast to the GC-based algorithms (e.g. [7,28]), which are not trivially extensible from binary to multi-label segmentation.

In our experiments, besides quantitative analysis involving current state-of-the-art techniques, we have also illu-

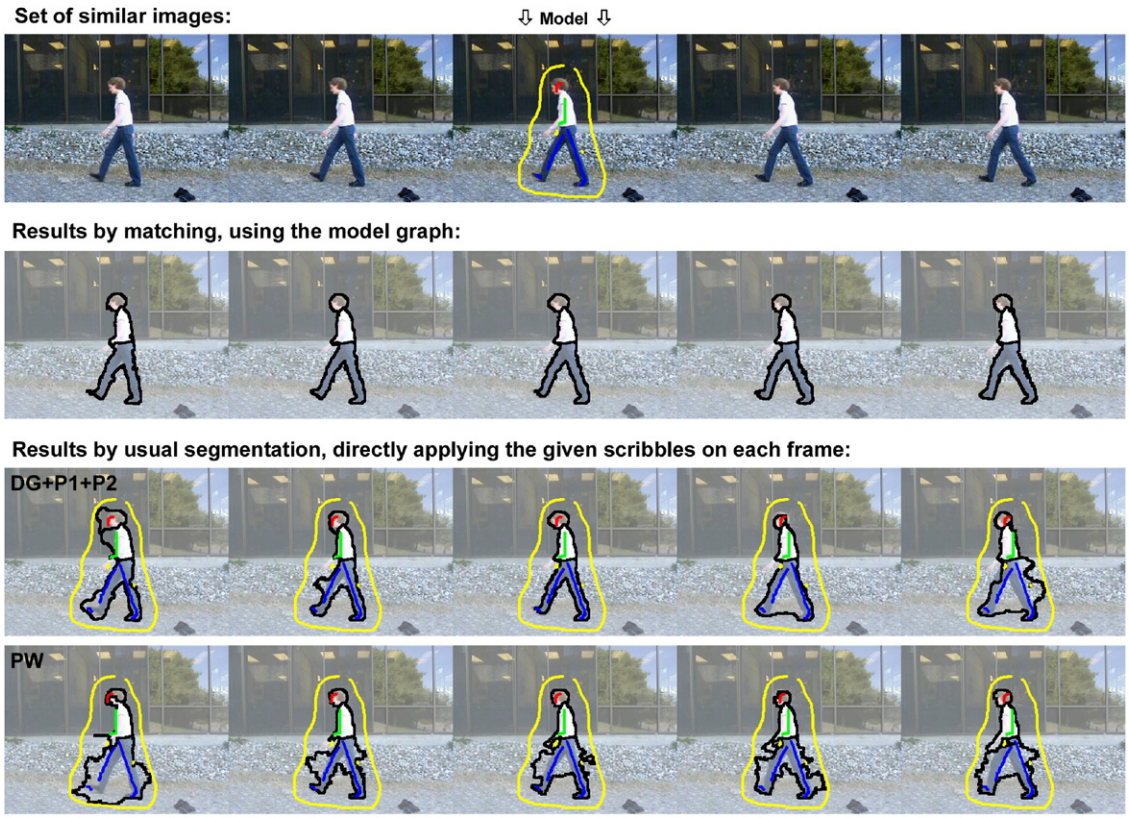


Fig. 26. First row: input images from the video sequence of a person walking straight [30], frames 63–67. Second row: segmentations by matching the corresponding attributed relational graphs by using frame 65 and its scribbles as the model graph. Third row: segmentations by using the usual interactive image segmentation and the same scribbles specifically designed for frame 65.

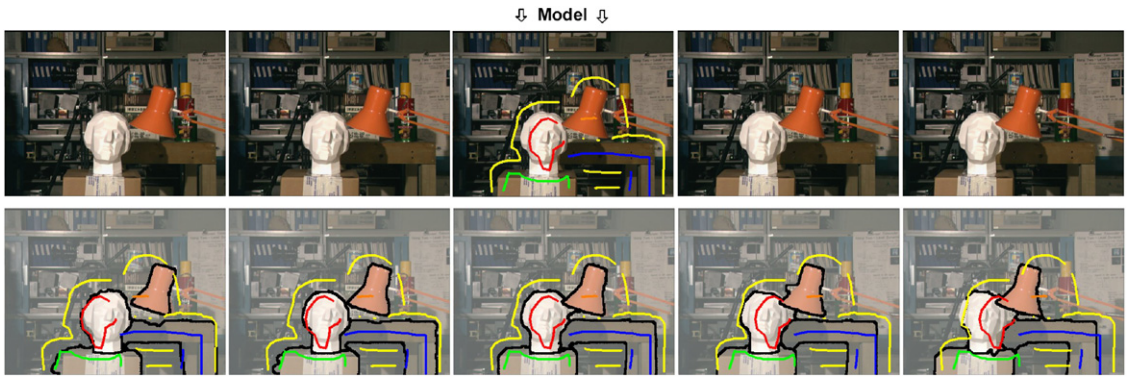


Fig. 27. First row: images provided by Dr. Y. Ohta and Dr. Y. Nakamura from University of Tsukuba. A single set of scribbles was specifically designed for the middle image, chosen as the model. No scribbles were defined on the other images of the sequence. Second row: for each image, the segmentation was obtained by matching the input attributed relational graph, built from this image, and the model graph. The single set of scribbles was placed over each segmentation result just to emphasize the object displacements.

strated the importance of the structural information by showing how λ_1 affects the performance when segmenting single images. The parameter λ_1 was used to balance the influence between color and spatial information on the segmentation result.

Moreover, we explored the re-usability of model graphs to segment multiple similar images, which can be applied to, for example, multi-label video segmentation based on models from key frames.

As a future step, further studies involving other graph matching techniques are necessary for more challenging instances involving segmentation of multiple similar images. In particular, the proposed algorithm is not suitable for very large displacements. For instance, in our experiments, good results were achieved when the objects were not very far from the single set of scribbles. More specifically, when each object was at least intersected by its correct scribble, limiting the displacements of the objects.

Another step would be to verify the possibility of automatically tuning λ_1 by using training data.

Acknowledgements

We are grateful to FAPESP, CAPES, CNPq, COFECUB and FINEP for their financial support. We thank Camille Couprie for providing the eroded seeds and an implementation for the accuracy

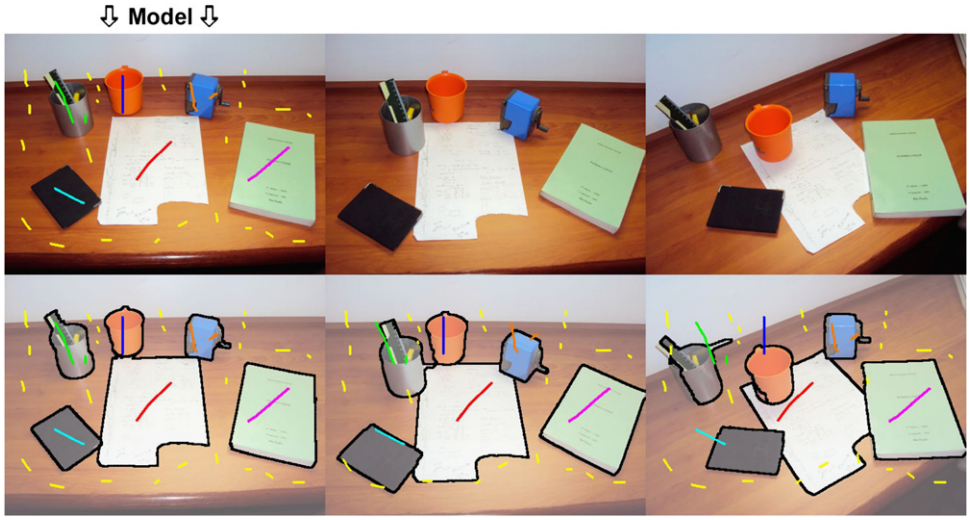


Fig. 28. Example with large changes on the object positions. A single set of scribbles was specifically designed for the image on the left side, chosen as the model. Each segmentation corresponds to the match between the model graph and each input graph representing each image. As in the previous figure, the single set of scribbles was placed over each segmentation result just to emphasize the object displacements.

Fig. 29. Example with large variability of the color information. Similarly to the former figure, the scribbles were drawn over the left image, chosen as the model. Each segmentation was obtained by matching the model graph with each input graph corresponding to each image.

indices used in [12], Kevin McGuinness for the context creator utility to load markers from image files and the implementation of the fuzzy boundary accuracy measure [23], Xue Bai for the input images from [3–5], Jue Wang for the executables for the Robust Matting [35], and all who made the source codes and databases freely available for our experiments. Finally, we would like to thank all the reviewers, which significantly contributed to improve the manuscript.

References

[1] T. Adamek, Using Contour Information and Segmentation for Object Registration, Modeling and Retrieval, Ph.D. Thesis, Dublin City University, 2006.
[2] R. Adams, L. Bischof, Seeded region growing, IEEE Transactions on Pattern Analysis and Machine Intelligence 16 (6) (1994) 641A–647.
[3] X. Bai, G. Sapiro, Distancecut: interactive segmentation and matting of images and videos, in: Proceedings of the International Conference on Image Processing, 2007.
[4] X. Bai, G. Sapiro, A geodesic framework for fast interactive image and video segmentation and matting, in: Proceedings of the IEEE International Conference on Computer Vision, 2007.
[5] X. Bai, G. Sapiro, Geodesic matting: a framework for fast interactive image and video segmentation and matting, International Journal of Computer Vision 82 (2) (2009) 113–132.
[6] S. Beucher, F. Meyer, The morphological approach to segmentation: the watershed transformation, in: E.R. Dougherty (Ed.), Mathematical Morphology in Image Processing, 1993, pp. 433–481.
[7] Y. Boykov, M. Jolly, Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images, Proceedings of the IEEE International Conference on Computer Vision, vol. 1, 2001, pp. 105–112.
[8] H. Bunke, Recent developments in graph matching, in: Proceedings of the International Conference on Pattern Recognition, 2000, pp. 2117–2124.
[10] R.M. Cesar Jr, E. Bengoetxea, I. Bloch, P. Larrañaga, Inexact graph matching for model-based recognition: evaluation and comparison of optimization algorithms, Pattern Recognition 38 (11) (2005) 2099–2113.
[11] L.A. Consularo, R.M. Cesar-Jr., I. Bloch, Structural image segmentation with interactive model generation, in: Proceedings of the IEEE International Conference on Image Processing, Piscataway, NJ, 2007.
[12] C. Couprie, L. Grady, L. Najman, H. Talbot, Power watershed: a unifying graph-based optimization framework, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (7) (2011) 1384–1399.

[13] J. Cui, Q. Yang, F. Wen, Q. Wu, C. Zhang, L.V. Gool, X. Tang, Transductive object cutout, in: IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.
[14] O. Duchenne, J.-Y. Audibert, R. Keriven, J. Ponce, F. Segonne, Segmentation by transduction, in: IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.
[15] G. Friedland, K. Jantz, R. Rojas, SIOX: simple interactive object extraction in still images, in: Proceedings of IEEE International Symposium on Multimedia, 2005, pp. 253–259.
[16] F. Ge, S. Wang, T. Liu, New benchmark for image segmentation evaluation, Journal of Electronic Imaging 16 (3) (2007).
[17] L. Grady, Random walks for image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (11) (2006).
[18] J. Guan, G. Qiu, Interactive image segmentation using optimization with statistical priors, in: ECCV International Workshop on the Representation and Use of Prior Knowledge in Vision, 2006.
[20] A. Levin, D. Lischinski, Y. Weiss, A closed-form solution to natural image matting, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (2) (2008) 228–242.
[21] Y. Li, J. Sun, C.-K. Tang, H.-Y. Shum, Lazy snapping, ACM Transactions on Graphics 23 (3) (2004).
[22] D. Martin, C. Fowlkes, D. Tal, J. Malik, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, Proceedings of the IEEE International Conference on Computer Vision, vol. 2, 2001, pp. 416–423.
[23] K. McGuinness, N.E. O'Connor, A comparative evaluation of interactive segmentation algorithms, Pattern Recognition 43 (2) (2010) 434–444.
[24] J. Ning, L. Zhang, D. Zhang, C. Wu, Interactive image segmentation by maximal similarity based region merging, Pattern Recognition 43 (2) (2010) 445–456.
[25] A. Noma, A. Pardo, R.M. Cesar-Jr, Structural matching of 2D electrophoresis gels using deformed graphs, Pattern Recognition Letters 32 (1) (2011) 3–11.
[26] B.L. Price, B. Morse, S. Cohen, Geodesic graph cut for interactive image segmentation, in: IEEE Conference on Comput Vision and Pattern Recognition, 2010.
[27] A. Protiere, G. Sapiro, Interactive image segmentation via adaptive weighted distances, IEEE Transactions on Image Processing 16 (4) (2007) 1046–1057.
[28] C. Rother, V. Kolmogorov, A. Blake, "GrabCut": interactive foreground extraction using iterated graph cuts, ACM Transactions on Graphics 3 (2004) 309–314.
[29] P. Salembier, L. Garrido, Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval, IEEE Transactions on Image Processing 9 (2000) 561–576.
[30] H. Sidenbladh, M.J. Black, D.J. Fleet, Stochastic tracking of 3D human figures using 2D image motion, in: Proceedings of the European Conference on Computer Vision, 2000, pp. 702–718.
[31] S. Todorovic, N. Ahuja, Region-based hierarchical image matching, International Journal of Computer Vision 78 (1) (2008) 47–66.
[32] S. Todorovic, N. Ahuja, Unsupervised category modeling, recognition, and segmentation in images, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (12) (2008) 2158–2174.
[33] W.H. Tsai, K.S. Fu, Error-correcting isomorphisms of attributed relational graphs for pattern analysis, IEEE Transactions on Systems, Man, and Cybernetics 9 (12) (1979) 757–768.
[34] L. Vincent, P. Soille, Watersheds in digital spaces: an efficient algorithm based on immersion simulations, IEEE Transactions on Pattern Analysis and Machine Intelligence 13 (6) (1991).
[35] J. Wang, M.F. Cohen, Optimized color sampling for robust matting, in: IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.
[36] R.C. Wilson, E.R. Hancock, Structural matching by discrete relaxation, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (6) (1997) 634–648.
[37] C. Yang, R. Duraiswami, N.A. Gumerov, L. Davis, Improved fast gauss transform and efficient kernel density estimation, in: Proceedings of the IEEE International Conference on Computer Vision, 2003.

Alexandre Noma received his M.Sc. (2003) and Ph.D. (2010) degrees in computer science from University of São Paulo, USP. He is currently pursuing a post-doctoral position at National Institute for Space Research at São José dos Campos, Brazil. His main research interests include graph matching, image segmentation and pattern recognition.

Ana B.V. Graciano received her M.Sc. (2007) in computer science from the University of São Paulo, USP. She is currently a Ph.D. candidate at USP, in collaboration with Télécom ParisTech, France. Her research interests are model-based object segmentation/classification, fusion of statistical and relational cues, graph-based pattern recognition, and medical imaging.

Roberto M. Cesar-Jr received his M.Sc. (1993, Electrical Engineering) from Unicamp and his Ph.D. (1997, Physics) from USP. He is currently a Full-Professor in the Department of Computer Science, IME-USP. He has experience in computer science, with emphasis on computer vision, pattern recognition, image processing and bioinformatics.

Luis Augusto Consularo is Systems Analyst at Informatics Department of Brazilian Superior Electoral Court, Brasília. He received his M.Sc. (1995) in computer science from Federal University of São Carlos, and his Ph.D. (2000) in computational physics from University of São Paulo. His research interests include computer vision and pattern recognition.

Isabelle Bloch is professor at Signal and Image Processing Department, Telecom ParisTech, CNRS LTCI, and coordinates the Image Processing and Understanding group. Her interests include 3-D image/object processing, computer vision, fuzzy and logics mathematical morphology, information fusion, fuzzy set theory, structural, graph-based and knowledge-based recognition, spatial reasoning, and medical imaging.

Signal, Image and Video Processing manuscript No.
(will be inserted by the editor)

Structural Segmentation and Labeling of Object Parts in Digital Video

Ana B. V. Graciano · Thiago M. Paixão · Roberto Hirata-Jr. · Roberto M. Cesar-Jr. · Isabelle Bloch

This is a submitted preprint.

Abstract This paper describes a model-based method for structural segmentation and labeling of object parts in digital video. The model consists of a set of object parts whose structural organization is defined interactively by a user upon a reference video frame. The model information is then applied to propagate the segmentation and labeling of the object parts in subsequent frames. The underlying data structure for this part-based model is an attributed relational graph (ARG) in which vertices encode object parts and edges represent the spatial arrangement of these parts. A backmapping method is proposed in order to register the position of each part with that of the model throughout time. Finally, segmentation and labeling of object parts are performed using an efficient graph-matching algorithm between the model graph and the graph issued from the frame under analysis. Experiments with real videos containing objects subject to different affine transformations are shown and discussed.

Keywords Structural Pattern Recognition · Object Part Segmentation and Labeling · Video Processing · Graph Matching · Backmapping

1 Introduction

Structural Pattern Recognition (SPR) concerns the description of patterns in terms of their composing parts and the relationships shared by these parts [22].

Ana B. V. Graciano, Thiago M. Paixão, Roberto Hirata-Jr., Roberto M. Cesar-Jr.
Institute of Mathematics and Statistics - IME
University of São Paulo - USP, Rua do Matão, 1010, 05508-090, São Paulo, Brazil
E-mail: {abvg,thiagopx,hirata,cesar}@ime.usp.br

Isabelle Bloch
TELECOM ParisTech, CNRS UMR 5141 LTCI
46 Rue Barrault, F-75634, Cedex 13, Paris, France
E-mail: isabelle.bloch@telecom-paristech.fr

Thanks to this characteristic, the SPR approach has been adopted in the solution of many computer vision applications, such as line-based pattern retrieval from large databases [16], face classification in video [17], medical imaging segmentation, image and video indexing, character recognition, among various others, as reported by Bunke et al. [3] and Conte et al. [11]. It is especially suited for tackling problems related to part-based detection and recognition [8, 12], besides applications focused on semantic interpretation of images, since it is possible to decompose a scene into multiple objects and describe relationships between/among them. For example, in a scene depicting people, the human body may be described in terms of its head, trunk and members (arms and legs). Each of these parts may be further decomposed into more detailed ones. An arm may be decomposed into (upper) arm, forearm and hand, which, in turn, may be decomposed as well, and so on. Yet, the main body parts are all related: each arm is *beside* the trunk, the head is *on top* of it, and the legs are *below* it, for a person standing up.

The mathematical concept of a graph is usually the underlying structure of choice for representing objects in this context. Structural graph-based methods have been proved suitable for segmenting and labeling object parts in static scenes [6], as well as in digital video processing [9,13–15,18], using techniques which combined appearance measures, object structure and even temporal features. Although these model-based video processing methods are capable of gathering different sources of information to describe an object, they usually present high computational complexity and expensive running times.

Because of the numerous semantics one may assign to what a target object is and how it is decomposed into parts, it is often necessary to define these concepts a priori by means of a model, a reference partition image, or user interaction. In particular, the interactive process allows the definition of what is to be segmented from the user's point of view, to produce segmentations that are in accordance with what a human would expect to achieve if he/she was to segment the video manually. Thus, interactive object segmentation methods are another class of algorithms that may serve the purpose of part-based segmentation and labeling. Examples of interactive video object segmentation techniques are described in [5] and in the successful Livecut [23] and Video SnapCut methods [1]. However, all these methods result in a binary partition of the video frames (background versus foreground), thus requiring modifications to produce part-based segmentations.

In this paper we address the problem of 2D part-based object segmentation and labeling in digital video using a new graph-based approach where object parts defined upon a single video frame by a user are segmented and labeled automatically in the other frames of the video by matching them to a model. This problem is analogous to multi-label interactive image segmentation [20]. The idea, however, is to propagate the model created by the user through frames of a video sequence, so that the object parts may be segmented and labeled in all frames without further user interaction. We adopt Attributed Relational Graphs (ARG) [25,26] as the underlying representation for both the model and the frame contents. The model is created interactively and de-

scribes the semantics of the objects of interest and their part decomposition, according to the user's point of view. The concept of object part is interpreted here as a union of 2D image regions organized according to some spatial configuration. This is an abstraction in the sense that a part may belong to a single object (e.g.: a human subdivided in head, trunk and legs) or represent a scene element of interest (e.g.: a scene with two "parts": a group of three men walking together versus the background).

Thus, a user first defines the target objects and their parts through a simple graphical interface, where he/she scribbles using a different color for each object part in a single reference frame of the video. This reference frame and the user traces are used to create a *model* ARG where each vertex represents an object part with associated features. Spatial relations are measured from these parts according to the image data and represented as attributed graph edges linking the corresponding vertices. A *frame* ARG is also automatically extracted from each video frame where the predefined object parts should be segmented and labeled. To avoid representing the entire frame, an approximate object detector is applied to limit the content in which to search for the target objects.

The problems of segmentation and labeling of object parts are formulated s an inexact graph matching problem. Thus, a solution is a correspondence between vertices from the frame ARG and those from the model ARG. However, because the appearance of an object and its parts typically change throughout a video, these changes must be considered in order to perform a reasonable matching between a given frame ARG and the model ARG. Two possible solutions may be adopted: either updating the model according to the frame space or mapping the moving object back to the model space. The former solution has been reported in [14] but errors in the update parameter estimation often led to model degeneration. To deal with this problem, we follow the second approach, i.e., input objects are mapped back to the model space immediately prior to recognition, an approach called *backmapping* here. This modification improved the stability of the results since the model parameters did not have to be re-estimated for each frame.

In this work, which extends the preliminary methodology discussed in [21], we apply the backmapping strategy before performing graph matching. This later step is then accomplished by means of an adaptation to video processing of the efficient graph matching optimization algorithm described in [20], which was originally applied to static images. This new optimization method has sped up the recognition step, while producing satisfactory results, since it accounts for the discrepancies between model and frame input throughout time.

This paper is structured as follows. Section 2 presents an overview of the proposed methodology main steps. Sections 3 through 4 describe the creation of model and frame graphs, while Section 5 details the target object segmentation process through graph matching. Section 6 discusses a set of experiments performed on sample videos from public datasets and self-acquired videos. Finally, Section 7 highlights some conclusions and ideas for future work.

2 Methodology overview

We present a methodology for part-based object segmentation in video that accounts for the structural arrangement of an object and its composing parts, as well as their appearance features.

The a priori knowledge of what an object is and how it is composed by its parts is introduced interactively by a user. The idea is to let the user identify in a single video frame, hereby called the *reference frame*, the objects of interest and their parts that are to be modeled. We call this process the *model creation* and this step essentially defines the target for the video segmentation that follows on. Although we could define the model by means of pre-established automatic rules (e.g. face detection rules for face tracking and recognition applications), the choice of creating a model interactively allows the segmentation and labeling of any kind of object, especially when the semantics of what is to be recognized is not well-defined, which is often the case [24] in general applications.

This interactively created model and the contents of the remaining video frames are represented by ARGs, i.e., graphs in which attribute vectors are assigned to vertices and to edges. Based on these representations, we then perform the automatic segmentation of the target objects throughout the input video.

Figure 1 provides an overview of the methodology. The chosen video reference frame (F_1) shows a boy on top of a skateboard. Suppose the relevant semantic for this scene according to a user implies the hair, face, trunk, legs/arms, and the skateboard itself. Then the user scribbles over these parts in the reference frame and this piece of information is used to generate a *model graph* (denote by G^M) henceforth simply referred to as *model*. This is an attributed relational graph where each vertex represents one of these parts, whereas edges connect neighboring parts.

Next, all remaining video frames F_i, $1 \leq i \leq n$ are automatically sequentially processed and a *frame graph* (denoted by G_i^I) is built from each of them. A frame graph is also an atributed relational graph, but vertices represent subsets of frame pixels, and edges connect sets containing neighbor pixels.

At each step, the model G^M is matched to each frame G_i^I thus resulting in a frame with regions labeled according to the target objects and their parts, denoted by R_i. The algorithm shown in Figure 2 summarizes these steps.

Firstly, the algorithm calls the procedure MODELGENERATE, which brings up the GUI used to define the target objects and their parts based on the reference frame F_1. The output of the procedure are the model graph G^M (Subsection 3.1), the set of control points (centroids of each part) C^M, and a region of interest around the modeled parts - the model silhouette mask $\texttt{silhouette}^M$. Next, APPROXIMATEFIND is called to define a limited region of interest in a given frame, the frame silhouette mask `silhouette`, which is input to INPUTARGGENERATE (Subsection 3.2). The result of the later is G_i^I, which is the frame graph for frame i. This ARG is backmapped to the model space by the procedure BACKMAP, resulting in the graph G_i^B (Section 4). Matching

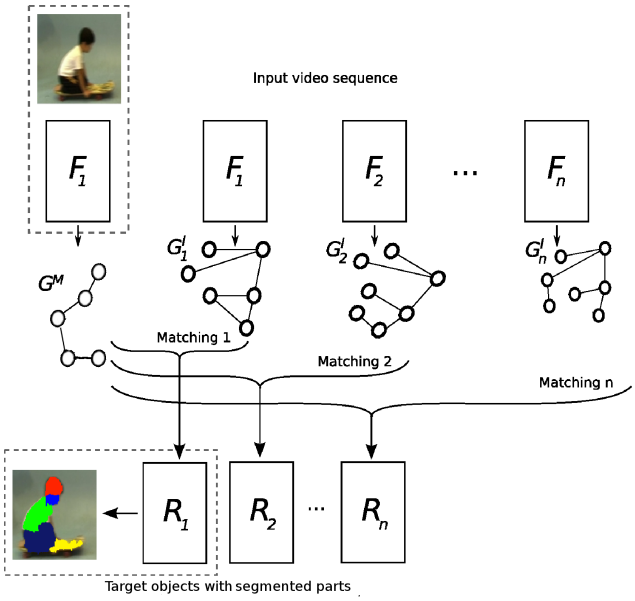


Fig. 1 Overview of the segmentation process. Each F_i represents an input frame from the video under analysis. At each step, a graph G_i^I is built for the current frame F_i, then it is transformed into the model space and finally matched with the model G^M. The respective resulting image is represented by R_i and contains the labeled and segmented object parts.

```
STRUCTURALSEGMENTATION(videoSequence = F_1, F_2, ..., F_n)
{G^M, C^M, silhouette^M} ← MODELGENERATE(F_1) ▷ See 3.1 for details.
T_i ← the identity affine transformation
for (i ← 1 to n)
    do
        silhouette ← APPROXIMATEFIND(F_i, T_i^{-1})
        ▷ See 3.2.1 for details.
        G_i^I ← INPUTARGGENERATE(F_i, silhouette)
        ▷ See 3.2 for details.
        G_i^B ← BACKMAP(G_i^I, T_i)
        ▷ See 4 for details.
        S_i ← GRAPHMATCH(G^M, G_i^B)
        {F^{S_i}, C_i^I} ← SEGMENT(F_i, S_i)
        ▷ See 5 for details.
        T_i ← CALCULATEBACKMAPPING(C^M, C_i^I)
```

Fig. 2 Pseudo-code for the structural matching process.

is then performed between G^M and G_i^B through the procedure GRAPHMATCH (Section 5), resulting in a set of pairs S_i that associate a label from an objectc part represented in G^M to a vertex in G_i^I. Then, the procedure SEGMENT (Section 5) creates a solution image F^{S_i} where the pixels that correspond to model objects are labeled according to S_i. A set of control points C_i^I is also issued

from the previous method, which are supplied to CALCULATEBACKMAPPING, together with the C^M, to compute the backmapping transformation that is used by APPROXIMATEFIND.

3 The model and frame graph representations

In this section, we detail the creation of the model graph that drives the labeling process, as well as the frame graphs.

Representation through attributed relational graphs. We adopt attributed relational graphs (ARG) to represent both the model and the frame graphs. An ARG is a graph defined by a quadruple $G = (V, E, \mu, \nu)$, where V denotes the set of vertices and E the set of edges. For all $v \in V$, there is an associated $p-$dimensional *object attribute vector* $\mu : V \to \mathbb{R}^p$. Analogously, for all $e \in E$, there is an associated $q-$dimensional *relational attribute vector* $\nu : E \to \mathbb{R}^q$. These vectors are responsible for adding relevant information to the graph data structure, since they may hold numerical or symbolic properties related to objects (vertices) and relations (edges) they are assigned to. This allows us to account not only for object structure but also for appearance and structural attributes.

Let G^M be the ARG defined by (V^M, E^M, μ^M, ν^M) and G_i^I the ARG defined by $(V_i^I, E_i^I, \mu_i^I, \nu_i^I)$.

In this paper, the attribute vectors μ^M and μ_i^I encode the following measures of the objects/regions associated to each vertex:

- the average of the color triple of an object/frame region;
- 2D coordinates of the object/frame centroids;
- a label identifying a model object (in the case of μ^M), or the model object associated to a frame region after the graph matching step (in the case of μ_i^I), which is the same for all vertices related to a single object.

On the other hand, the attribute vectors ν^M and ν_i^I encode the geometric distance vector between the centroids of both objects/frame regions associated to the vertices they connect. These geometric vectors are all normalized with respect to the maximum geometric vector in terms of modulus. This normalization factor corresponds to the largest scalar distance between centroids of adjacent objects/regions.

3.1 Model graph: MODELGENERATE(F_1)

The model G^M is generated interactively only once by following the next steps.

First, the user must define what are the target objects and their parts. In order to avoid complex or tedious work, the interaction process is limited to a few steps performed over a given reference frame of the input video. These steps should be easy for the user and yet able to produce a labeling

of all regions belonging to those objects, a process known as the semantic step [5]. To accomplish this, we first ask the user to scribble over regions of the reference frame belonging to each object (or its parts) of interest, as well as over background regions. For each object or object part, a scribble is drawn as a line using a unique color that serves as an identifying label for this respective target object.

The method does not require a precise definition of the borders of the objects nor of their parts. Nevertheless, the scribbles defining an object must be contained in the set of pixels that depicts this object in the reference frame. Figure 3(a) shows the scribbles drawn by the user to segment the skateboard boy example shown in Figure 1.

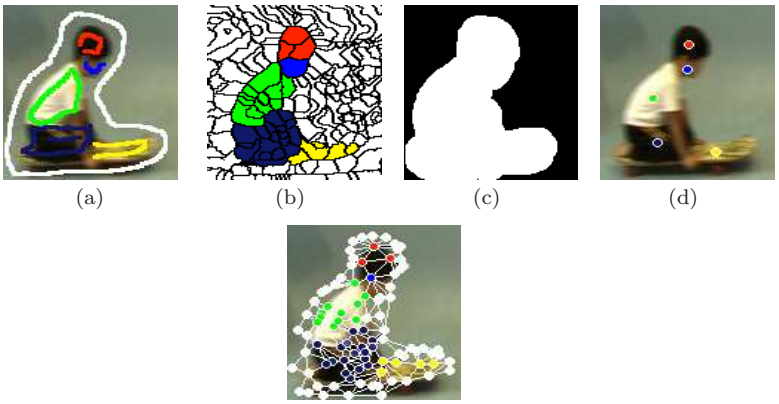


Fig. 3 Model creation steps: (a) labeled scribbles drawn by the user; (b) watershed segmentation with the basins labeled according to the user scribbles; (c) ROI defined as the dilated object silhouette; (d) centroids of object parts used for motion estimation; (e) frame graph computed inside an initial rectangular ROI.

The next step in model creation is to expand these scribbles to the whole extent of each target object, without further effort from the user. In order to do this, we use the user-defined scribbles (Figure 3) as markers to the Beucher-Meyer morphological segmentation paradigm [2,19]. This paradigm consists in applying the watershed transform [27] to the morphological gradient of the reference frame constrained by a marker image. This image is obtained from a homotopic filtering of the original gradient with respect to an image containing only the user scribbles.

Figure 3(b) shows the result of the watershed transform, where black lines depict watershed basin limits. Then, the basins of the watershed transform are labeled according to the scribbles used as markers in the morphological segmentation paradigm (Figure 3(b)).

Every region marked as background takes a single label and the set of background regions is considered as one object. Later, these labels are used by the segmentation algorithm to identify objects in each video frame.

After label expansion, we define a region of interest (ROI) around the union of all regions labeled as an object after the morphological segmentation. Then a morphological dilation is applied to enlarge the ROI and allows for some background regions to be included. The size and shape of the structuring element may depend on the application and on the video itself, thus being set in an *ad-hoc* manner. Figure 3(c) shows the ROI obtained for the skateboard boy example. This ROI is useful as a frame blob in which one can find the objects defined by the user. It is also used to compute a set C^M of centroids of non-background regions, which are taken as control points to later estimate object motion during video processing. Figure 3(d) shows the computed centroids for the labeled parts of Figure 3(b). [1]

Finally, the ARG G^M may be computed. Each labeled region inside the ROI gives rise to a vertex in V^M, whereas an edge is included in E^M to connect vertices corresponding to adjacent regions. The vertex and edge attribute vectors previously defined are measured from each region and their relations according to the reference frame. All vertices are labeled with the color of their originating basin, which correspond to the colors of the original scribbles. Figure 3(e) shows the model graph for the given reference frame.

It is worth noticing that, if a non marker-based watershed is adopted and it produces an over-segmentation of the image, some image filtering may be required for some applications to reduce the final number of regions.

3.2 Frame graph creation

There are two important steps to create a frame graph G_i^I:

1. find an approximate location of the object in the current frame and derive a ROI for this frame;
2. apply the watershed algorithm to the ROI and generate the corresponding ARG.

The first step aims to reduce the regions of a frame to those where one will potentially find the target objects. The second one aims to partition this ROI into regions whose boundaries are likely to contain or to merge into the true boundaries of the target objects. These are covered in more detail in the remainder of this subsection.

3.2.1 Approximate object detection: APPROXIMATEFIND(F_i, T^{-1})

In order to avoid processing the whole frame data at each time, a ROI limits the region in which to search for target objects prior to segmentation and labeling. This step is called approximate object detection.

[1] In the worst case, the ROI might cover almost the whole frame content.

The approximate object detection relies on a few hypotheses. First, we suppose that the video rate is high enough so that changes between two consecutive frames are not significant. In this case, it is possible to reuse information about the location of an object from a frame to the next one and to obtain coherent recognition results throughout time. Furthermore, we adopt an affine transformation as the motion model. This is a simple yet effective model to applications such as digital surveillance with fixed cameras. Nevertheless, any other type of transformation could be adopted, without modifying the remainder of the method.

In practice, this step consists in adapting a mask from frame F_{i-1} to frame F_i, according to the segmentation obtained for F_{i-1} and the updated control points that locate each model object in this frame.

This step is accomplished by transforming the original dilated model ROI S (obtained during model creation, Section 3.1) according to an affine transformation T_{i-1} that maps a labeled object in F_{i-1} onto the corresponding object in the reference frame. T_{i-1}^{-1} denotes the inverse of T_{i-1}. Then, $T_{i-1}^{-1}(S)$ denotes the region of interest of frame F_{i-1}. The ROI is dilated in order to account for the possible inaccuracy of T_{i-1} applied to F_i due to the movement of objects between subsequent frames. This approach also avoids constructing an ROI only from the frame information, which could suffer from noise. Updating a ROI from frame to frame guarantees a better spatial consistency and robustness with respect to noise. Figure 4 presents a diagram of this process.

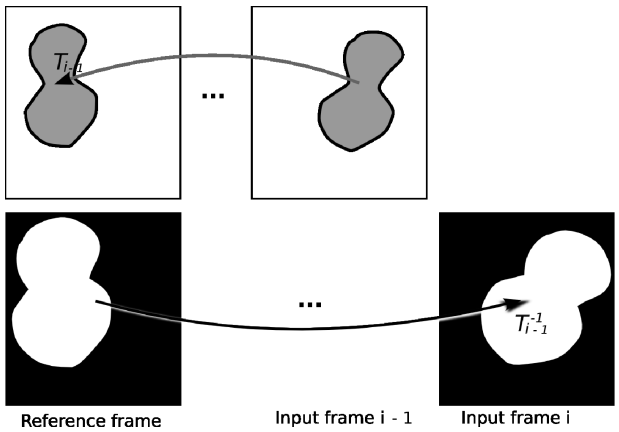


Fig. 4 Approximate object detection. In the first row, the transformation T_{i-1} maps the object of input frame $i-1$ onto the object of the reference frame. In the second row, T_{i-1}^{-1} is used to map the dilated reference frame ROI onto input frame i, thus providing an approximate object location for the subsequent frame.

The transformation T_{i-1} can only be estimated after recognition of frame F_{i-1} because it provides control points that match those computed during

model creation and allows us to calculate a proper affine transformation, as explained in Section 4. Also, because of the soft motion hypothesis, T_{i-1} should be reasonably correct.

3.2.2 Frame graph from watershed

After limiting the content in frame F_i according to the computed ROI, the ARG G_i^I can be created. For this, the watershed algorithm is applied only to the intersection between $T_{i-1}^{-1}(S)$ and frame F_i^I. This algorithm produces an over-segmentation of the input frame under the ROI. Each of these over-segmented regions generates a vertex and its corresponding attributes, whereas adjacent regions generate edges and their attributes. All computed attributes are the same as those present in the model graph. Although this over-segmentation might look like a drawback at first, this rough segmentation frequently retains the true borders of the target objects that might be under the ROI. Thus, at this point, these target objects are not yet labeled, nor entirely segmented as they should. The task of merging the over-segmented regions into wider regions corresponding to target objects is left to the matching algorithm, which will be explained in Section 5.

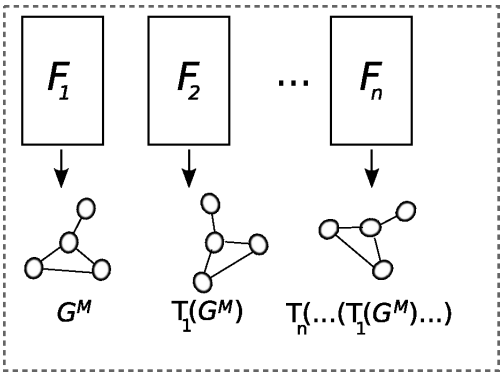
4 Backmapping: BACKMAP(G_i^I, T_i) and CALCULATEBACKMAPPING(C^M, C_i^I)

The model ARG parameters (vertex and edge attributes) are calculated based on the reference frame. As the scene changes throughout the video, the target objects may also change, often presenting attributes that differ from those defined in the beginning. For instance, objects may shift, rotate or present changes in scale, color, or suffer distortions from lighting effects.

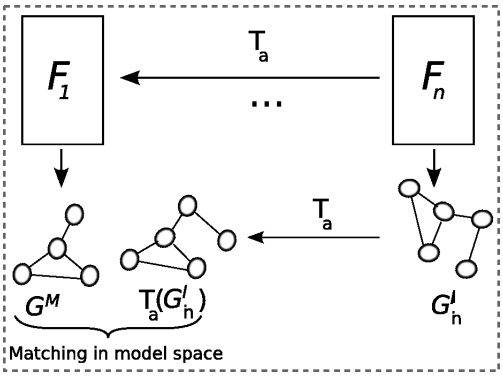
Because the matching procedure is not invariant to such changes, it is necessary to normalize one of the graphs. In terms of scene geometry, a possibility is to update the coordinates of each object in the model by applying an affine transformation after each processed frame, as proposed in [14]. However, experiments with this method have shown that the errors accumulated by successive applications of the estimated affine transformation may lead to model degeneration and the process has to be re-initialized by the user with a new model.

Another possibility is to reverse this procedure by mapping the current frame information back to the model space before the matching procedure [7], as illustrated in Figure 5. This backmapping idea has been successfully employed in the system hereby described.

In this work, backmapping is defined as the task of estimating an affine transformation that maps a frame graph G_i^I back to the model space, i.e., a graph G_i^B. This procedure and the watershed step are strongly related. The watershed is applied to a limited ROI and it partitions the image into various regions, i.e., watershed basins. We assume that an object part is a set of



(a) Model update.



(b) Backmapping.

Fig. 5 Model update versus backmapping. In the model update approach, the model undergoes sequential transformations that may degenerate the model. In the second case, the model is fixed and each frame graph is backmapped to the model space.

connected basins. Certain basins will be part of the object and others will not. Also, according to the video dynamics, some basins related to the same object part remain connected and others do not. We therefore filter the resulting image by selecting the largest connected component for each part.

Therefore, we need to estimate the backmapping matrix, which is the task of procedure CALCULATEBACKMAPPING(C^M, C_i^I). The equation below defines an affine transformation:

$$\overrightarrow{q} = \delta(A\overrightarrow{s} + \overrightarrow{b}) \tag{1}$$

where A is a 2×2 non-singular linear matrix, $\overrightarrow{b}$ is a 2×1 real-valued vector defining a translation, δ is any real scalar value. In order to calculate the backmapping matrix, we assign to $\overrightarrow{s}$ the averages of the centroid coordinates of all regions associated to a single vertex in G^M, represented by the set of control points C^M. Thus each value assigned to $\overrightarrow{s}$ corresponds to the centroid of a specific object identified by a given label. We assign to $\overrightarrow{q}$ the correspond-

ing centroid coordinates (for a given label) obtained from the object parts segmented in a given frame, represented by the set of control points C_i^I.

A linear system is built from these correspondences and the parameters of the backmapping matrix are computed by solving this system. Because an affine transformation is given by six parameters, there should be at least three parts in a given segmented input frame and also in the reference frame. In general, there are more than three control points in each frame to be mapped, thus the system is over-determined and an approximate solution, in the least-mean squares sense, is calculated.

5 Labeling and segmentation through graph matching: GRAPHMATCH(G^M, G_i^B)

The labeling problem consists in assigning a given set of labels of the target object to each region in the constrained input frame, according to the compatibility between these regions and the set of modeled objects. A solution to this labeling problem is a mapping from the transformed set of vertices V_i^I (backmapped frame graph G_i^B) to the set V^M (graph model) [14]. Because $|V_i^I|$ is usually larger than $|V^M|$ due to the over-segmentation phenomenon, many possible mappings between G_i^I and G^M exist. Thus, a solution is actually a desirable many-to-one mapping between V_i^I and V^M, which corresponds to merging frame regions belonging to the same object. This task is carried out by defining a cost function to evaluate the quality of each possible match. The cost function should then be minimized and, hence, the graph matching process is expressed as an optimization problem. This class of mappings is called inexact graph matching.

Two aspects should be addressed: the optimization algorithm and the cost function. Various graph matching algorithms have been proposed in the literature [4,10,28]. Here we adopt the matching algorithm described in [20]. When matching G_i^I with G^M, it is important to take into account that these graphs usually have very different structures. This is due to the fact that vertices of G^M are associated to object parts defined by the user whereas those of G^I relate to regions issued from the watershed algorithm, typically over-segmented.

In order to cope with this problem, the authors of [20] propose the use of an additional auxiliary structure called the *deformation graph* $G_i^D = (V_i^D, E_i^D, \mu_i^D, \nu_i^D)$. This structure is initialized as a copy of the model graph, and *deformed* by the assignment of an input node $v_i^I \in V_i^I$ to $v_i^D \in V_i^D$. This assignment affects v_i^D and a set of edges connected to it, since the attributes of vertex v_i^D are computed as a blend between the attributes of the corresponding original model vertex and those from the frame vertex being assigned to v_i^D.

The deformation graph is compared to the original model by means of a cost function which measures the dissimilarity between the deformed and the original vertex and edge attributes respectively. This allows measuring the impact of assigning a given frame vertex to a certain model vertex by computing the structural and appearance deformations derived from this mapping. Note

that the "deformation" applies to assignments and attributes, but does not affect the graph topology. The comparison is performed independently for each vertex $v_i^I \in V_i^I$.

Then, the optimization algorithm evaluates the cost function for each vertex of the frame graph and assigns to it the model label of the object which minimizes the cost. Let $v^M \in V^M$ and $e^M \in E^M$ be a vertex and an edge of G^M, respectively. The cost function, denoted by f, is a weighted sum of object and structural properties, being defined in our experiments as:

$$f(G_i^D, G^M) = \alpha c_V(v_i^D, v^M) + \frac{(1-\alpha)}{|E_i^D(v_i^D)|} \sum_{e \in E_i^D(v_i^D)} c_E(e, e^M) \tag{2}$$

where c_V and c_E are dissimilarity measures between vertices and edges, respectively. Thus, compatible pairs of vertices or edges present small dissimilarity value and contribute to minimizing f.

In order to compute the deformation between the object attributes of v_i^D and v^M, the term $c_V(v_i^D, v^M)$ is defined as a distance measure between the attribute vectors of v_i^D and v^M.

Thus, the object dissimilarity measure computes the difference between the vertices attributes that characterize the deformed and model vertices.

Similarly, if $e \in E_i^D(v_i^D)$ and $e^M \in E^M$ is its corresponding model edge, then $c_E(e, e^M)$ is a measure of the deformation between both edges. For instance, we adopt the measure defined in [6], where the edges attribute is a vector $\nu(e)$ connecting the centers of gravity of the regions linked by this edge:

$$c_E(e, e^M) = \gamma_E \frac{|\cos(\theta) - 1|}{2} + (1 - \gamma_E)|\|\nu(e)\| - \|\nu(e^M)\|| \tag{3}$$

The value θ is the angle between $\nu(e)$ and $\nu(e^M)$, whereas the parameter γ_E, $0 \leq \gamma_E \leq 1$, controls the weights of the modular and angular dissimilarities between vectors connecting two given regions. Thus, the total impact caused on the edges directly connected by v_i^D is computed as the modular and angular differences between the relational attribute vectors of each pair (e, e^M). The parameter γ_E may be defined empirically according to the relevance assigned to the modulus or the angle.

Finally, the output of the GRAPHMATCH algorithm is a mapping S_i from the labels of vertices in G^M to the vertices of G_i^B. Because each frame graph vertex is associated to a region in the respective original input frame F_i, procedure SEGMENT(F_i, S_i) replaces the pixels of those regions by their corresponding model labels given by S_i. This results in a new labeled image F^{S_i}, where regions have been merged into parts of interest and segmented. Besides this, a set of control points C_i^I is recomputed based upon the labeled regions for this frame. The new centroids correspond to the average of the centroids of all regions assigned to the same label. These control points are then used as input to the backmapping procedure, as explained in Section 4.

6 Experimental results

In this section, we report a series of experiments performed to evaluate the methodology. The previously discussed algorithms were implemented in C++ using the Intel OpenCV library. The experimental platform consisted of a 1.73GHz Celeron machine with 1GB of RAM. The dataset includes three 320 × 240 MPEG sequences with a frame-rate of 30fps. The adopted values for the parameters α (weight for vertex cost, see Section 5) and δ (weight for modulus edge cost) are summarized in Table 1.

Sample frames from each video are presented with recognized parts superimposed on the original sequence [2]. In all three cases, model creation took about 5 seconds, including graph creation based on user input and on the scribble-based watershed. The approximate time to process each video frame was 4 seconds, which is acceptable since the experiments did not aim at real time processing.

Table 1 Summary of the performed experiments.

Video name	Total number of frames	α	γ
Skateboard boy	36	0.3	0.7
Doll	60	0.3	0.7
CAVIAR project	988	0.3	0.7

(a) (b) (c)

Fig. 6 Scribbles drawn by users, defining object models for the three videos.

Experiment 1: translation and constant motion.

In the first sequence, a boy performs a translation at nearly constant speed from left to right. The labeling of the boy's body, the skateboard and some background are shown in Figure 6(a). The result (Figure 6) is presented by 8 video frames sampled at intervals of 8 frames, from a total of 36. From visual inspection, results obtained for this video were satisfying, with very

[2] Results available on `www.vision.ime.usp.br/~abvg/videos/segmentation/`

small variations along the frames due to excessive blurring and low resolution images.

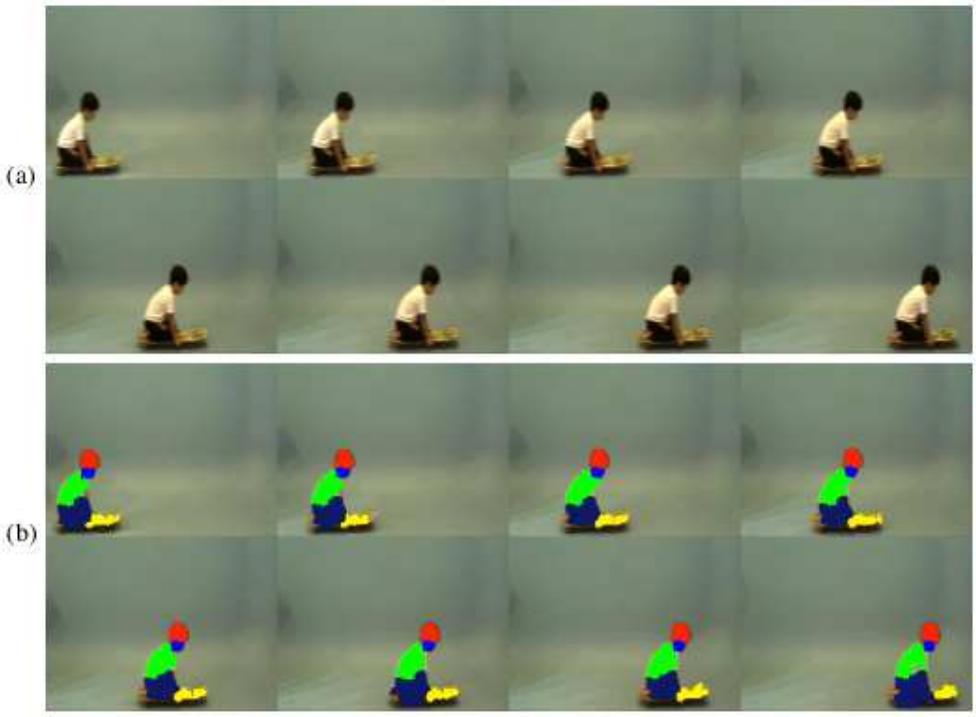


Fig. 7 Results for the video "Skateboard boy": (a) original sequence; (b) sample frames with recognized parts.

Experiment 2: rotation.

The second scenario is a synthetic rotation of a doll. Figure 6(b) shows the scribbles for both object and background. Due to the flat background, the watershed gives rise to a single background catchment basin. This may mislead the recognition step because the whole background is represented by a single vertex. If this vertex is misclassified, the whole background is misclassified as well. Thus, new vertices are generated around the object by splitting the background into a $p \times q$-rectangular grid ($p = q = 25$ pixels in this sequence). The result (Figure 8) shows 8 video frames sampled in intervals of 8 frames, from a total of 60 frames (rotation of 60 degrees).

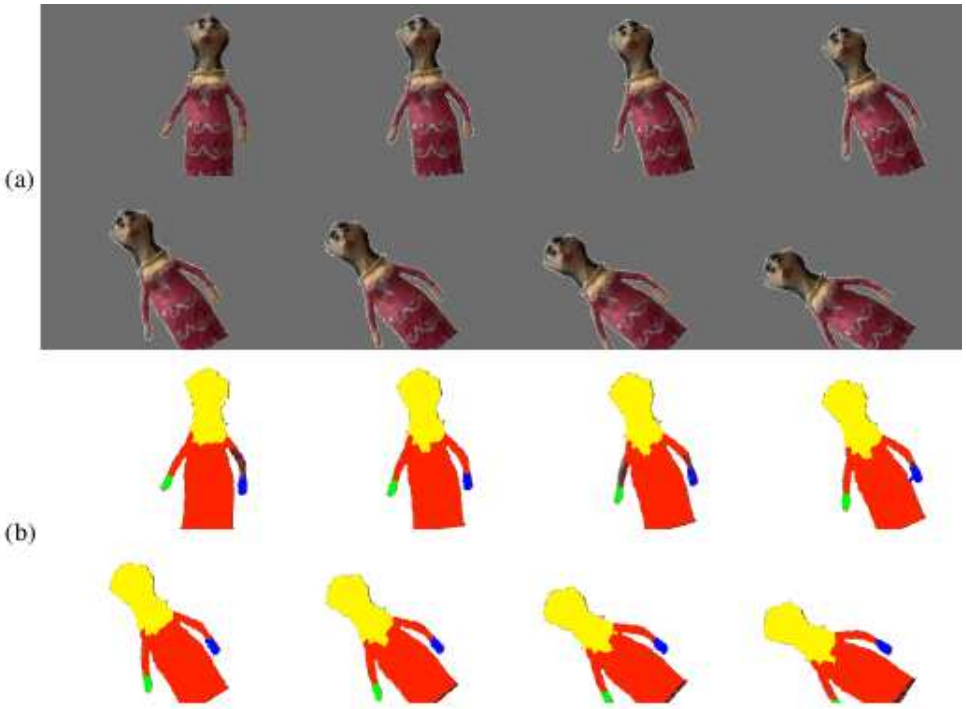


Fig. 8 Results of doll's sequence. (a) original sequence; (b) sample frames with recognized parts.

Experiment 3: deformable objects in motion. The third video is part of the test case scenarios of the CAVIAR[3] project. In this video, three men are walking in the corridor of a shopping mall. In a 2D view, this movement can be described as an object translation from top to bottom, with some scaling effect. This scenario presents some challenging elements, such as other moving people, shadows, heterogeneous background and poor object resolution in the reference image and subsequent frames.

The three men were modeled as a set of three tops (head and trunk)and three bottoms (legs and feet). Figure 6(c) depicts the model of object and background which guided the recognition process. Figure 9 depicts the result of object recognition along the video frames. Our approximate object detection strategy eliminates non-target objects, performing the role of dynamic background subtraction. In certain frames, dark strips on the floor were merged with the dark pants of the three targets due to the fusion of regions by the watershed algorithm. Those regions represent different elements in the scenario but have similar appearance. In spite of this error, the advantage of the structural approach is that artifacts with similar properties as those of the model do

[3] EC Funded CAVIAR project/IST 2001 37540, found at URL: http://homepages.inf.ed.ac.uk/rbf/CAVIAR/

not introduce recognition errors if they are not too close to the target objects. This is not the case with techniques that rely exclusively on object appearance for instance.

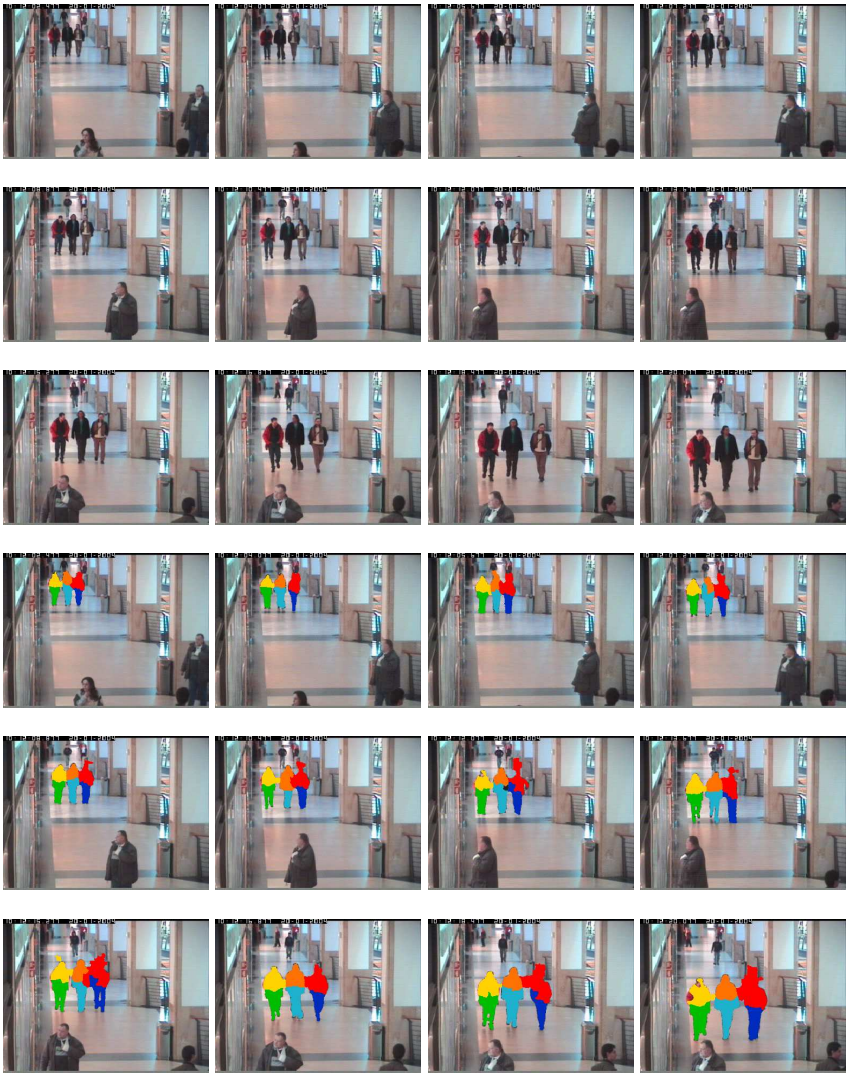


Fig. 9 Results of CAVIAR sequence. Top three rows: frames from the original sequence; last three rows: corresponding frames with segmented parts.

Besides this, an additional experiment is reported. Because different parts are segmented and labeled, the video may be selectively edited based on the segmented parts. Figure 10 shows an application of the methodology in which the body of one of the three target men of the CAVIAR sequence has been

erased, preserving but the head in the resulting edited video. This was achieved by applying the recognition process to all frames and then replacing the pixels labeled as the man's trunk and legs automatically by the corresponding scene background (estimated on-the-fly from the video). Although the visual effect is not perfect because of the differences implied by the estimated background, the example shows that the labeled regions may be correctly selected and modified. This test demonstrates how the methodology may be applied for video editing purposes such as chroma-key, in which the erased parts could have been subject to some special effect or replaced by a different texture.

Fig. 10 An application of the methodology: digital chroma-key or automatic object removal. The body of the man on the left has been erased from the sequence simply by removing the parts labeled as his legs and trunk, after the segmentation process.

Comparative assessment: Model update versus backmapping. An experiment comparing the backmapping approach with the model update method described in [14] was also performed. The aim was to analyze how well both approaches were able to correctly segment the modeled objects in each frame and follow their movement. For this, we have first sampled the CAVIAR video by selecting frames at intervals of 20 frames from beginning to end. Then we manually annotated the centroid coordinates of each target part. The coordinates obtained for each part were interpolated in order to build a ground truth of their expected trajectories. Next, the video was processed using two model variants: one which relied on backmapping to relate the model with the unknown input yet to be classified, and another which adopts the model update based on consecutive classifications to predict where the object might be next.

The model creation and matching steps previously discussed in this paper were applied in both cases. The centroid coordinates of each part were obtained automatically and their measured trajectories have been plotted together with the ground truth as shown in Figures 11(a) and 11(b). The graphs show that

the backmapping approach produces more stable classifications throughout time, even if it is not precisely aligned with the ground truth. This is either due to the approximate nature of the ground truth itself, or due to errors in the classification process. Yet, the curves are close to the ideal solution and parts are not lost in the segmentation of the video. On the other hand, the model update diverges faster and is more error-prone, since it is not possible to recover the model once it has been degraded.

7 Concluding remarks

Video analysis is a very important research topic, especially nowadays, when various devices produce a huge amount of pictorial data. Human resources alone are unable to analyze all these data and new automatic ways to do it are called for. Recent advances in video analysis show that Structural Pattern Recognition is a promising way to model objects, their parts and their interrelationships, as well as to recognize them in a given image.

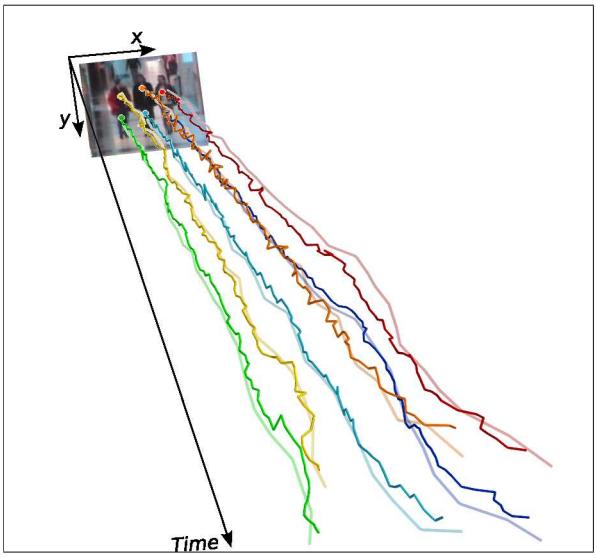
In this paper, we proposed a methodology for structural object segmentation and labeling in video based on an interactively generated model. Once a user defines the semantics of the target objects, the system analyzes each frame of a digital video automatically by creating an attributed relational graph for each frame and matching it to the model, thus achieving the segmentation and labeling of the object parts of interest.

To cope with object changes throughout the video, we adopted a backmapping approach, which maps the input data back to the model space. In comparison with the reverse approach of model update, which recomputes the model at each frame transition, the first solution improved the robustness of the segmentation process, avoiding problems such as quick model degeneration.

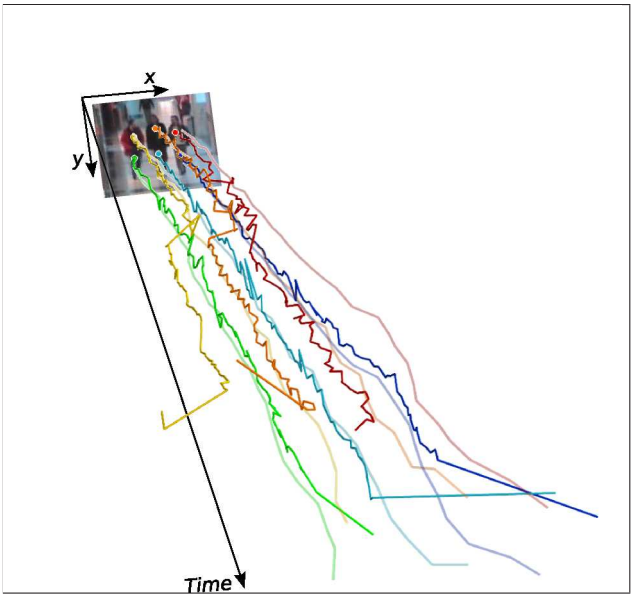
The human-computer interaction required to define the model is simple, since it relies only on scribbles drawn on a reference image. However, it enables the acquisition of an informative model that corresponds to what the user expects to recognize. Finally, the adoption of a new graph matching algorithm decreased considerably the complexity of this step in comparison with the original methodology [14], resulting in a more efficient system without compromising the accuracy of part segmentation and recognition. In practice, when tested against the goose[4] sequence presented in [14], it has taken 5 minutes to process the set of 410 frames, with visually similar results, whereas the other method reported 78 minutes.

Acknowledgements The authors are grateful to FAPESP, CNPq, CAPES, FINEP and COFECUB for their financial support.

[4] Results available on www.vision.ime.usp.br/~abvg/videos/segmentation2/



(a)



(b)

Fig. 11 Part-based object trajectories: ground truth (darker lines) versus trajectories of the segmented parts (softer lines) using (a) backmapping and (b) model update.

References

1. Bai, X., Wang, J., Simons, D., Sapiro, G.: Video SnapCut: robust video object cutout using localized classifiers. In: ACM SIGGRAPH, pp. 1–11 (2009)

2. Beucher, S., Meyer, F.: Mathematical morphology in image processing, chap. The morphological approach to segmentation: the watershed transformation. Marcel Dekker (1992)
3. Bunke, H.: Recent developments in graph matching. In: Int. Conf. Pattern Recognition, pp. 2117–2124 (2000)
4. Bunke, H., Allermann, G.: Inexact graph matching for structural pattern recognition. Pattern Recognition Letters **1**(4), 245–253 (1983)
5. Castagno, R., Ebrahimi, T., Kunt, M.: Video segmentation based on multiple features for interactive multimedia applications. IEEE Transactions on Circuits and Systems for Video Technology **8**(5), 562 –571 (1998)
6. Cesar-Jr, R.M., Bengoetxea, E., Larranaga, P., Bloch, I.: Inexact graph matching for model-based recognition: evaluation and comparison of optimization algorithms. Pattern Recognition **38**(11), 2099–2113 (2005)
7. Chai, J., Xiao, J., Hodgins, J.: Vision-based control of 3d facial animation. In: ACM SIGGRAPH/Eurographics Symp. Comput. Animation, pp. 193–206 (2003)
8. Chen, Y., Zhu, L., Lin, C., Yuille, A., Zhang, H.: Rapid inference on a novel and/or graph for object detection, segmentation and parsing. In: J. Platt, D. Koller, Y. Singer, S. Roweis (eds.) Advances in Neural Inf. Processing Syst., pp. 289–296. MIT Press, Cambridge, MA (2008)
9. Conte, D., Foggia, P., Jolion, J.M., Vento, M.: A graph-based, multi-resolution algorithm for tracking objects in presence of occlusions. Pattern Recognition **39**(4), 562 – 572 (2006)
10. Conte, D., Foggia, P., Sansone, C., Vento, M.: Graph matching applications in pattern recognition and image processing. In: IEEE Int. Conf. Image Processing, vol. 2, pp. 21–24 (2003)
11. Conte, D., Foggia, P., Sansone, C., Vento, M.: Thirty years of graph matching in pattern recognition. Int. J. Pattern Recognition and Artificial Intell. **18**(3), 265–298 (2004)
12. Felzenszwalb, P.F., Schwartz, J.D.: Hierarchical matching of deformable shapes. In: IEEE Int. Conf. Comput. Vision and Pattern Recognition, pp. 1–8 (2007)
13. Gomila, C., Meyer, F.: Graph-based object tracking. In: IEEE Int. Conf. Image Processing, vol. 2, pp. 41–44 (2003)
14. Graciano, A.B.V., Cesar-Jr, R.M., Bloch, I.: Graph-based object tracking using structural pattern recognition. In: Brazilian Symp. Comput. Graph. and Image Processing, pp. 179–186 (2007)
15. Grundmann, M., Kwatra, V., Han, M., Essa, I.A.: Efficient hierarchical graph-based video segmentation. In: IEEE Int. Conf. Comput. Vision and Pattern Recognition, pp. 2141–2148 (2010)
16. Huet, B., Hancock, E.: Relational object recognition from large structural libraries. Pattern Recognition **35**(9), 1895–1915 (2002)
17. Kokiopoulou, E., Frossard, P.: Graph-based classification of multiple observation sets. Pattern Recognition **43**(12), 3988 – 3997 (2010)
18. Lee, J., Oh, J., Hwang, S.: Clustering of video objects by graph matching. In: IEEE Int. Conf. Multimedia and Expo, pp. 394 –397 (2005)
19. Meyer, F.: Color image segmentation. In: IEEE Int. Conf. Image Processing, pp. 303–306 (1992)
20. Noma, A., Graciano, A.B.V., Cesar-Jr, R.M., Consularo, L.A., Bloch, I.: Interactive image segmentation by matching attributed relational graphs. Pattern Recognition **45**(3), 1159 – 1179 (2012)
21. Paixao, T.M., Graciano, A.B.V., Cesar-Jr, R.M., Hirata-Jr, R.: A backmapping approach for graph-based object tracking. In: Brazilian Symp. Comput. Graph. and Image Processing, pp. 45–52 (2008)
22. Pavlidis, T.: Algorithms for shape analysis of contours and waveforms. IEEE Trans. Pattern Anal. Mach. Intell. **2**(4), 301–312 (1980)
23. Price, B.L., Morse, B.S., Cohen, S.: Livecut: Learning-based interactive video segmentation by evaluation of multiple propagated cues. In: Int. Conf. Comput. Vision, pp. 779 – 786 (2009)
24. Qiu, G., Yuen, P.C.: Interactive imaging and vision – ideas, algorithms and applications. Pattern Recognition **43**(2), 431 – 433 (2010)

25. Schalkoff, R.J.: Pattern recognition: statistical, structural and neural approaches. John Wiley & Sons, Inc., New York, NY, USA (1991)
26. Tsai, W.H., Fu, K.S.: Error-correcting isomorphisms of attributed relational graphs for pattern analysis. IEEE Trans. Systems, Man and Cybernetics **9**(12), 757 –768 (1979)
27. Vincent, L., Soille, P.: Watersheds in digital spaces: an efficient algorithm based on immersion simulations. IEEE Trans. Pattern Anal. Mach. Intell. **13**(6), 583–598 (1991)
28. Wilson, R.C., Hancock, E.R.: Structural matching by discrete relaxation. IEEE Trans. Pattern Anal. Mach. Intell. **19**, 634–648 (1997)

ÍNDICE REMISSIVO

Estatísticas consecutivas, 75

Grafo, 9
- aleatório, 13
- aleatório genérico, 13
- casamento entre, 11, 31
- clique, 10
- de entrada, 11
- de observação, 29, 67
- definição, 9
- dirigido ou dígrafo, 9
- estatístico-relacional, 5, 22, 70
- modelo, 11
- não-dirigido, 9
- ordem, 9
- relacional com atributos, 10, 29
- relacional generalizado com atributos, 14
- tamanho, 9

Imagem
- bidimensional, 20
- de entrada, 27
- partição, 20
- tridimensional, 20

Objeto
- estruturado, 2, 19
  - modelo computacional, 22
  - modelo conceitual, 21
- sistema estruturado, 2, 19
  - modelagem, 63

Overflow, 75

Reconhecimento de objetos, 1
- estatístico-estrutural, 25, 30, 75
  - aplicações, 59
- paradigma estatístico, 2
- paradigma estrutural, 2

Região
- expandida, 21
- refinada, 20

Segmentação
- completa, 20

Underflow, 78

REFERÊNCIAS BIBLIOGRÁFICAS

[Aur95] Aurélio. *Dicionário Aurélio da Língua Portuguesa*. Ed. Nova Fronteira, 1995.

[Bar05] D. Bartz. Virtual endoscopy in research and clinical practice. *Comput. Graph. Forum*, 24(1):111–126, 2005.

[BCCJ06] I. Bloch, O. Colliot e R. M. Cesar-Jr. On the ternary spatial relation "between". *IEEE Trans. Syst. Man Cybern., Part B Cybern.*, 36(2):312–327, 2006.

[BIN05] H. Bunke, C. Irniger e M. Neuhaus. Graph matching: Challenges and potential solutions. Em *Proc. Int. Conf. Image Analysis and Process.*, páginas 1–10, 2005.

[BRB08] S. Banik, R.M. Rangayyan e G.S. Boag. Landmarking of computed tomographic images to assist in segmentation of abdominal tumors caused by neuroblastoma. Em *30th IEEE Int. Conf. Eng. Medicine and Biology Soc.*, páginas 3126–3129, 2008.

[BVZ01] Y. Boykov, O. Veksler e R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1222–1239, 2001.

[CCP10] P. Campadelli, E. Casiraghi e S. Pratissoli. A segmentation framework for abdominal organs from CT scans. *Artificial Intell. in Medicine*, 50(1):3–11, 2010.

[CFH05] D. Crandall, P. Felzenszwalb e D. Huttenlocher. Spatial priors for part-based recognition using statistical models. Em *Proc. IEEE Comp. Soc. Conf. Comput. Vision and Pattern Recognition*, volume 1, páginas 10–17. IEEE Computer Society, 2005.

[CFSV04] D. Conte, P. Foggia, C. Sansone e M. Vento. Thirty years of graph matching in pattern recognition. *Int. J. Pattern Recognition and Artificial Intell.*, 18(3):265–298, 2004.

[CGX08] K. Cheng, L. Gu e J. Xu. A novel shape prior based level set method for liver segmentation from mr images. Em *Int. Conf. Tech. Applications in Biomedicine*, páginas 144–147, 2008.

[CH93] D. K. Y. Chiu e G. Harauz. A method for inferring probabilistic consensus structure with applications to molecular sequence data. *Pattern Recognition*, 26(4):643–654, April 1993.

[CJBBL05] R. M. Cesar-Jr., E. Bengoetxea, I. Bloch e P. Larrañaga. Inexact graph matching for model-based recognition: Evaluation and comparison of optimization algorithms. *Pattern Recognition*, 38(11):2099–2113, 2005.

[DHS00] R. O. Duda, P. E. Hart e D. G. Stork. *Pattern Classification*. Wiley-Interscience Publication, 2000.

[FGV00] P. Foggia, R. Genna e M. Vento. Prototype learning with attributed relational graphs. Em *Proc. Joint IAPR Int. Workshops on Advances in Pattern Recognition*, páginas 447–456, London, UK, 2000. Springer-Verlag.

[FH05] P. Felzenszwalb e D. P. Huttenlocher. Pictorial structures for object recognition. *Int. J. Comput. Vision*, 61(1):55–79, 2005.

[FMA95] FMA. Foundational model of anatomy ontology, 1995. `http://fma.biostr.washington.edu`.

[GCBE98] G. Le Goualher, D. L. Collins, C. Barillot e A. C. Evans. Automatic identification of cortical sulci using a 3d probabilistic atlas. Em *Proc. Int. Conf. Medical Image Computing and Comput. Assisted Intervention*, páginas 509–518, 1998.

[Gra07] A. B. V. Graciano. Rastreamento de objetos baseado em reconhecimento estrutural de padrões. Dissertação de Mestrado, Institute of Mathematics and Statistics - University of São Paulo, March 2007.

[GT07] L. Getoor e B. Taskar, editors. *Introduction to Statistical Relational Learning*. MIT Press, 2007.

[GW93] R. C. Gonzalez e R. E. Woods. *Digital Image Processing*. Addison-Wesley Publishing Company, 1a. edição, 1993.

[Jö06] P. S. Jörgensen. Segmentation of male abdominal fat using MRI. Dissertação de Mestrado, Informatics and Mathematical Modelling, Technical University of Denmark, 2006.

[JDM00] A. K. Jain, R. Duin e J. Mao. Statistical pattern recognition: a review. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(1):4–37, 2000.

[JNF+07] F. Jäger, L. G. Nyúl, B. B. Frericks, F. K. Wacker e J. Hornegger. Whole body MRI intensity standardization. Em *Bildverarbeitung fü die Medizin*, páginas 459–463, 2007.

[JZ97] A. Jain e D. Zongker. Feature selection: Evaluation, application, and small sample performance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(2):153–158, 1997.

[Kar90] N. Karssemeijer. A statistical method for automatic labeling of tissues in medical images. *Mach. Vision Appl.*, 3:75–86, April 1990.

[LHS08] Rajasvaran Logeswaran, Tan Wooi Haw e Shakowat Zaman Sarker. Liver isolation in abdominal MRI. *J. Med. Syst.*, 32(4):259–268, 2008.

[Li09] Stan Z. Li. *Markov random field modeling in image analysis*. Springer, 2009.

[Mar04] D. J. Marchette. *Random Graphs for Statistical Pattern Recognition*. Wiley-Interscience, 2004.

[MK06] S. Muraki e Y. Kita. A survey of medical applications of 3d image analysis and computer graphics. *Syst. Comput. Japan*, 37(1):13–46, 2006.

[NB04] M. Neuhaus e H. Bunke. A probabilistic approach to learning costs for graph edit distance. Em *Proc. Int. Conf. Pattern Recognition*, páginas 389–393, 2004. vol. 3.

[Neu04] A. Neumaier. Clouds, fuzzy sets and probability intervals. *Reliable Computing*, 10:249–272, 2004.

[NGCJ+12] A. Noma, A. B. V. Graciano, R. M. Cesar-Jr., L. A. Consularo e I. Bloch. Interactive image segmentation by matching attributed relational graphs. *Pattern Recognition*, 45(3):1159 – 1179, 2012.

[Nom10] A. Noma. *Duas abordagens para casamento de padrÃţes de pontos usando relaÃğÃţes espaciais e casamento entre grafos*. Tese de Doutorado, Institute of Mathematics and Statistics - University of São Paulo, July 2010.

[NvBF+02] W. J. Niessen, C. M. van Bemmel, A. F. Frangi, M. J. A. Siers e O. Wink. Model-based segmentation of cardiac and vascular images. Em *Proc. IEEE Int. Symp. Biomedical Imaging*, páginas 22–25, 2002.

[Ope10] OpenGalen. The galen common reference model, 2010. `http://www.opengalen.org/`.

[OS01] S. D. Olabarriaga e A. W. M. Smeulders. Interaction in the segmentation of medical images: a survey. *Medical Image Anal.*, 5-2:127 – 142, 2001.

[Pai10] T. M. PaixÃčo. Rastreamento de objetos utilizando reconhecimento estrutural de padrÃţes: uma abordagem com modelo estÃątico. Dissertação de Mestrado, Institute of Mathematics and Statistics - University of São Paulo, 2010.

[Pav80] T. Pavlidis. *Structural Pattern Recognition*. Springer-Verlag, 1980.

[PBM03] H. Park, P. H. Bland e C. R. Meyer. Construction of an abdominal probabilistic atlas and its application in segmentation. *IEEE Trans. Med. Imaging*, 22(4):483–492, 2003.

[Pel90] E. Peli. Contrast in complex images. *J. Opt. Soc. Am. A*, 7(10):2032–2040, Oct 1990.

[Per98] P. Perez. Markov random fields and images. *CWI Quarterly*, 11(4):413–437, 1998.

[PGCJHJ08] T. M. PaixÃčo, A. B. V. Graciano, R. M. Cesar-Jr. e R. Hirata-Jr. A backmapping approach for graph-based object tracking. Em *Proc. Brazilian Symp. Comput. Graph. and Image Process.*, páginas 45–52, Washington, DC, USA, 2008. IEEE Computer Society.

[SAS03] F. Serratosa, R. Alquezar e A. Sanfeliu. Function-described graphs for modelling objects represented by sets of attributed graphs. *Pattern Recognition*, 36(3):781–798, March 2003.

[SB04] N. D. Singpurwalla e J. M. Booker. Membership functions and probability measures of fuzzy sets. *J. of the American Statistical Assoc.*, 99:867–877, 2004.

[Sch92] R. J. Schalkoff. *Pattern Recognition: statistical, structural and neural approaches*. John Wiley & Sons, 1992.

[SHB08] M. Sonka, V. Hlavac e R. Boyle. *Image Processing, Analysis, and Machine Vision*. CENGAGE Learning, 3 edição, 2008.

[SW85] M. Spann e R. Wilson. A quad-tree approach to image segmentation which combines statistical and spatial information. *Pattern Recognition*, 18(3âĂŞ4):257 – 269, 1985.

[TF79] W. H. Tsai e K. S. Fu. Error-correcting isomorphisms of attributed relational graphs for pattern analysis. *IEEE Trans. Syst. Man Cybern., Part C Appl. Rev.*, 9(12):757–768, 1979.

[Val03] J. Valentin, editor. *Annals of the ICRP. Basic Anatomical and Physiological Data for Use in Radiological Protection: Reference Values*. Elsevier, 2003.

[Var02] L. R. Varshney. Abdominal organ segmentation in CT scan images: A survey. 2002.

[VS91] L. Vincent e P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(6):583–598, 1991.

[WFM+05] S. Wolfsberger, M. Forster, L. Mroz, R. Wegenkittl e K. Buhler. Advanced virtual endoscopic pituitary surgery. *IEEE Trans. Visualiz. and Comput. Graph.*, 11(5):497–507, 2005. Student member - A. Neubauer.

[WH97] R. C. Wilson e E. R. Hancock. Structural matching by discrete relaxation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(6):634–648, 1997.

[WPNK08] J. Wu, S. Poehlman, M. D. Noseworthy e M. V. Kamath. Texture feature based automated seeded region growing in abdominal MRI segmentation. Em *Proc. Int. Conf. BioMedical Engineering and Informatics*, páginas 263–267, Washington, DC, USA, 2008. IEEE Computer Society.

[WS90] A. K. C. Wong e H. C. Shen. Random graphs. *Syntactic and Structural Pattern Recognition: Theory and Applications*, 1990.

[WY85] A. K. C. Wong e M. You. Entropy and distance of random graphs with application to structural pattern recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 7(5):599–609, September 1985.

[ZB07] Y. Zhou e J. Bai. Multiple abdominal organ segmentation: An atlas-based fuzzy connectedness approach. *IEEE Trans. Information Tech. in Biomedicine*, 11(3):348–352, 2007.