Segmentação de imagens e validação de classes
por abordagem estocástica

Leandro Cavaleri Gerhardinger

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito: 14/03/2006

Assinatura:_____

Segmentação de imagens e validação de classes por abordagem estocástica

Leandro Cavaleri Gerhardinger

Orientador: *Prof. Dr. João do Espírito Santo Batista Neto*

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos necessários para obtenção do título de Mestre em Ciências de Computação e Matemática Computacional.

USP – São Carlos
Março/2006

Agradecimentos

Aos meus pais Leopoldo e Heloisa, aos meus irmãos Leopoldo, Rodrigo, Helena e Elisa e à minha companheira Camila, por todo amor, carinho e suporte, dados a mim incondicionalmente.

Aos meus amigos da república, do laboratório e da turma, por toda amizade e companhia que deixaram meu dia-a-dia mais alegre e interessante.

Ao professor João do Espírito Santo Batista Neto, sem o qual este trabalho não seria possível, por tantos anos de orientação com muita inteligência e disposição.

Ao sempre disponível professor Mário de Castro, pela indispensável ajuda no desenvolvimento deste trabalho.

A todos os professores e funcionários deste Instituto, onde passei esta tão importante fase da minha vida.

À CAPES, pelo apoio financeiro.

Resumo

Uma etapa de suma importância na análise automática de imagens é a segmentação, que procura dividir uma imagem em regiões cujos pixels exibem um certo grau de similaridade. Uma característica que provê similaridade entre pixels de uma mesma região é a textura, formada geralmente pela combinação aleatória de suas intensidades. Muitos trabalhos vêm sendo realizados com o intuito de estudar técnicas não-supervisionadas de segmentação de imagens por modelos estocásticos, definindo texturas como campos aleatórios de Markov. Um método com esta abordagem que se destaca é o EM/MPM, um algoritmo iterativo que combina a técnica EM para realizar uma estimação de parâmetros por máxima verossimilhança com a MPM, utilizada para segmentação pela minimização do número de pixels erroneamente classificados.

Este trabalho desenvolveu um estudo sobre a modelagem e a implementação do algoritmo EM/MPM, juntamente com sua abordagem multiresolução. Foram propostas uma estimação inicial de parâmetros por limiarização e uma combinação com o algoritmo de *annealing*. Foi feito também um estudo acerca da validação de classes, ou seja, a busca pelo número de regiões diferentes na imagem, mostrando as principais técnicas encontradas na literatura e propondo uma nova abordagem, baseada na distribuição dos níveis de cinza das classes. Por fim, foi desenvolvida uma extensão do modelo para a segmentação de malhas em duas e três dimensões.

Abstract

An important stage of the automatic image analysis process is segmentation, that aims to split an image into regions whose pixels exhibit a certain degree of similarity. Texture is known as an efficient feature that provides enough discriminant power to differentiate pixels from distinct regions. It is usually defined as a random combination of pixel intensities. A considerable amount of researches has been done on non-supervised techniques for image segmentation based on stochastic models, in which texture is defined as Markov Random Fields. Such an important method in this category is the EM/MPM, an iterative algorithm that combines the maximum-likelihood parameter estimation model EM with the MPM segmentation algorithm, whose aim is to minimize the number of misclassified pixels in the image.

This work has carried out a study on stochastic models for segmentation and shows an implementation for the EM/MPM algorithm, together with a multiresolution approach. A new threshold-based scheme for the estimation of initial parameters for the EM/MPM model has been proposed. This work also shows how to incorporate the concept of annealing to the current EM/MPM algorithm in order to improve segmentation. Additionally, a study on the class validity problem (search for the correct number of classes) has been done, showing the most important techniques available in the literature. As a consequence, a gray level distribution-based approach has been devised. Finally, the work shows an extension of the traditional EM/MPM technique for segmenting 2D and 3D meshes.

Sumário

Sumário	vii
1 Introdução	1
1.1 Objetivos	2
1.2 Organização	2
2 Segmentação por Modelos Estocásticos	3
2.1 Segmentação EM/MPM	4
2.1.1 Modelo para o Campo de Rótulos	5
2.1.2 Modelo para a Imagem Observada	7
2.1.3 Algoritmo MPM	7
2.1.4 Algoritmo EM	11
2.1.5 Algoritmo EM/MPM	12
2.2 Segmentação EM/MPM Multiresolução	13
2.2.1 Modelagem Multiresolução	15
2.2.2 Algoritmo EM/MPM Multiresolução	15
2.3 Considerações	16
3 Implementação	17
3.1 Algoritmo Detalhado	17
3.2 Estimação Inicial de Parâmetros	20
3.3 <i>Annealing</i>	22
4 Validação de Classes	25
4.1 Critérios de Validação	25
4.2 Distância entre Distribuições	31
4.2.1 Estatística de Kolmogorov-Smirnov	31
4.2.2 Distância de Bhattacharyya	37
4.3 Considerações	38
5 Resultados	41
5.1 Imagens Sintéticas	41
5.2 Imagens Médicas	47
5.3 Cenas Naturais	49
5.4 Malhas	53
6 Conclusões e Trabalhos Futuros	59
Referências Bibliográficas	61

Introdução

A **segmentação** é tida como uma das principais etapas nas aplicações de análise de imagem e de maior complexidade computacional. Seu objetivo se baseia na separação das regiões de uma imagem que são associadas aos objetos nela representados. Fugindo da subjetividade da interpretação, a segmentação concentra-se nas propriedades de descontinuidade e similaridade destas regiões, tais como nível de cinza, borda ou textura. Desta forma, o espaço de busca por objetos numa etapa posterior é reduzido consideravelmente.

Muitas abordagens para segmentação podem ser encontradas na literatura, sendo normalmente classificadas em três grupos básicos [Gon87]: limiarização (*thresholding*), segmentação baseada em bordas e segmentação baseada em regiões. Recentemente várias outras categorias têm sido levadas em consideração tais como as técnicas baseadas em cores [GBS90], redes neurais e lógica fuzzy [BP92] e algoritmos genéticos [AFDV90, ABML89, ABP90].

Uma das mais importantes propriedades a serem analisadas na segmentação é a textura, característica quantitativa que, embora seja uma idéia intuitiva para a percepção humana, não possui uma fácil definição para a área de processamento de imagens. Neste contexto, ela pode ser entendida como um atributo formado por padrões aleatórios, periódicos ou determinísticos, que representam o arranjo espacial dos tons de cinza numa dada região da imagem [Jai89, IEE90, Jah97].

Métodos de segmentação que utilizam análise de texturas são normalmente classificados como baseados em região, podendo ser divididos em três categorias [GDO85, RB93]: métodos estatísticos, estruturais e baseados em modelos. Os métodos estatísticos demandam uma etapa de extração de características e, por isso, são muito custosos computacionalmente [Son99]. Já os métodos estruturais são utilizados para texturas de comportamento bem definido (determinístico), não sendo adequados, portanto, para grande parte das imagens do mundo real. Por fim, os métodos baseados em modelos podem ser utilizados em uma grande variedade de tipos de texturas, podendo ser classificados em [RB93] modelos fractais e estocásticos. Estes últimos, em particular, vêm sendo cada vez mais objeto de estudo devido aos resultados bastante promissores obtidos.

O processo de segmentação pode ser **supervisionado** ou **não-supervisionado**. No primeiro, há uma etapa anterior à da segmentação que visa obter um conjunto de informações a partir de uma coleção de dados (por exemplo, um banco de imagens). Apesar

de tentadora, a idéia de obter-se características sobre a imagem antes de segmentá-la nem sempre é possível e conveniente. Já no processo não-supervisionado, essas informações devem ser estimadas durante o processo de segmentação. Essa é uma abordagem bastante positiva especialmente em sistemas de visão computacional, nos quais a interação com o usuário é indesejável.

Um método não-supervisionado de segmentação de imagens por textura baseado em um modelo estocástico é o EM/MPM, proposto por Comer e Delp [CD00]. Apesar dos poucos estudos realizados com o uso desta técnica, ela se mostrou muito adequada, sobretudo na segmentação de imagens em níveis de cinza formadas por microtexturas, já que o modelo utilizado é baseado em um campo aleatório de Markov (CAM, também encontrado na literatura por MRF).

1.1 Objetivos

Este trabalho tem como objetivo estudar e implementar o método EM/MPM de segmentação de imagens, bem como modelar o seu uso em uma abordagem multiresolução. A fim de reduzir o número de parâmetros a serem previamente definidos pelo usuário e melhorar os resultados da segmentação, neste trabalho também é proposta uma técnica para a estimação inicial de parâmetros por limiarização e um esquema para combinar o método com o conceito de *annealing* (resfriamento simulado).

Visando encontrar o número correto de regiões diferentes dentro de uma imagem, é feito um estudo sobre as principais técnicas de validação de classes aplicadas às técnicas não-supervisionadas de segmentação baseadas em modelos estocásticos. Ao final deste estudo, é proposta uma abordagem específica para o método EM/MPM, que busca analisar as funções de distribuição do modelo.

Por fim, neste trabalho são apresentados os resultados obtidos com diferentes tipos de imagem, sendo também proposta uma extensão do método implementado para a segmentação de malhas de duas e três dimensões.

1.2 Organização

Esta dissertação está dividida em seis capítulos. Já no capítulo 2 são apresentadas a modelagem do método de segmentação de imagens EM/MPM e sua abordagem multiresolução. A implementação do algoritmo EM/MPM proposto, juntamente com as técnicas de estimação inicial de parâmetros por limiarização e de *annealing* são discutidas no capítulo 3. O estudo da validação de classes e do comportamento de seus critérios são apresentados no capítulo 4.

Os resultados dos métodos implementados aplicados em diferentes tipos de imagem, assim como a modelagem e da extensão da segmentação EM/MPM para aplicação em malhas são abordadas no capítulo 5. Por fim, no capítulo 6, são apresentadas as considerações finais do trabalho, juntamente com as sugestões para trabalhos futuros.

Segmentação por Modelos Estocásticos

Os objetos visualizados em imagens do mundo real são, em sua grande maioria, caracterizados por microtexturas de comportamento não determinístico, ou seja, aleatório. Nesses casos, portanto, justifica-se a utilização de uma abordagem probabilística de segmentação. Diversos estudos sobre modelos estocásticos de imagem vêm sendo desenvolvidos [GG84, DE87, GP97, CD00] e aplicados com sucesso em diversas situações [RGR97, Bal03, BTTA05, SP05].

De modo geral, os algoritmos estatísticos de segmentação definem os rótulos dos pixels de uma imagem como variáveis aleatórias que formam um campo aleatório bidimensional. Existem muitos modelos para campos aleatórios [Ros97], tais como o de Gibbs, de Ising, de Potts, de Pickard e de Markov. Este último, mais conhecido como CAM (Campo Aleatório Markoviano - *Markovian Random Field*), é bastante adequado para a modelagem de microtexturas, pois define a função de probabilidade do campo por meio de características puramente locais, ou seja, características de vizinhança. Além disso, os CAM's podem representar praticamente qualquer problema de contexto estocástico, modelando características complexas com o uso de relativamente poucos parâmetros.

Em uma imagem digital, os objetos são geralmente percebidos como um agrupamento de pixels em uma dada região. A representação de cada objeto pode ser feita rotulando-os e construindo um **campo de rótulos**. Desta forma, cada pixel da imagem original corresponde a apenas uma **classe** de objetos. Entretanto, esta é exatamente a informação buscada na segmentação. No modelo estocástico, a imagem é um **dado observado** e o campo de rótulos é um **dado faltante** (ou **ausente**). Este é, portanto, um problema da família dos **problemas de dados incompletos**. Neste caso, a tarefa de segmentação é, na verdade, uma tarefa de otimização estatística de um dado critério.

Muitos algoritmos que aproximam a estimativa do máximo da distribuição *a posteriori* (MAP - *Maximum a Posteriori*) do campo de rótulos, dado a imagem observada, foram propostos [DE87, LD89]. Mais recentemente, em outros trabalhos [MMP87, CD00] é estudado o critério conhecido como **maximizador das marginais a posteriori** (MPM - *Maximizer of the Posterior Marginals*). Este critério tem se mostrado mais apropriado para a segmentação de imagens [MMP87], já que o MAP associa o mesmo custo para toda segmentação incorreta, enquanto o MPM impõe esse custo baseando-se no número de pixels classificados incorretamente.

Na segmentação não-supervisionada, os parâmetros do modelo estocástico utilizado são desconhecidos, sendo necessário estimá-los. Alguns métodos conhecidos com esta finalidade são: **SEM** (*Stochastic Expectation-Maximization*), **ICE** (*Iterative Conditional Estimator*) e **EM** (*Expectation-Maximization*). Assim, métodos estocásticos não-supervisionados de segmentação de imagens podem ser desenvolvidos por meio da combinação de algoritmos de estimação de parâmetros da distribuição da imagem e otimização estatística para a segmentação, como ICE/MPM [GP97] e EM/MPM [CD00].

Os métodos estocásticos de segmentação estudados nesse trabalho foram o EM/MPM e uma variação deste que utiliza a abordagem multiresolução. Este capítulo descreverá seus respectivos modelos nas seções 2.1 e 2.2.

2.1 Segmentação EM/MPM

O método estocástico não-supervisionado de segmentação de imagens EM/MPM foi originalmente proposto por Mary L. Comer e Eduard J. Delp [CD95, CD00]. Como seu nome diz, ele é a combinação de dois algoritmos: EM para estimação de parâmetros da distribuição da imagem e MPM para otimização estatística do campo de rótulos (segmentação).

Para modelar o problema, serão utilizadas as seguintes notações: $\mathbf{X}$ é o campo de rótulos e $\mathbf{Y}$ a imagem observada, ambos definidos como campos aleatórios no látice retangular S , com um número N de elementos (neste caso, pixels); X_s e Y_s são variáveis aleatórias de $\mathbf{X}$ e $\mathbf{Y}$ na posição $s \in S$; L é o número de rótulos supostamente conhecido; $\mathbf{x} = (x_1, \dots, x_N)$ e $\mathbf{y} = (y_1, \dots, y_N)$ são realizações de $\mathbf{X} = (X_1, \dots, X_N)$ e $\mathbf{Y} = (Y_1, \dots, Y_N)$, respectivamente; $\Omega_{\mathbf{X}}$ e $\Omega_{\mathbf{Y}}$ são os espaços das possíveis realizações de $\mathbf{X}$ e $\mathbf{Y}$, respectivamente, onde $x_s \in \{1, \dots, L\}$ e $y_s \in \{0, \dots, 255\}$ para qualquer $s \in S$.

Assim como outros métodos estocásticos, a modelagem utilizada para a resolução do problema de segmentação está adequada à formulação como um problema de dados incompletos. Neste caso, a imagem original é o dado observado e a imagem segmentada (campo de rótulos) é o dado ausente. Segundo o teorema de Bayes, a probabilidade condicional $p_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta})$ do campo de rótulos dada a imagem original e o conjunto de parâmetros $\boldsymbol{\theta}$ pode ser escrita como¹

$$p_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta}) = \frac{p_{\mathbf{X}}(\mathbf{x}) \times f_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta})}{f_{\mathbf{Y}}(\mathbf{y}|\boldsymbol{\theta})}, \quad (2.1)$$

onde a função $p_{\mathbf{X}|\mathbf{Y}}$ é chamada de **distribuição de probabilidade *a posteriori***, enquanto $p_{\mathbf{X}}$ é denominada **distribuição de probabilidade *a priori***. A função $p_{\mathbf{X}|\mathbf{Y}}$ é a probabilidade do campo de rótulos dada a imagem observada, $p_{\mathbf{X}}$ é a distribuição do campo de rótulos e $f_{\mathbf{Y}|\mathbf{X}}$ é a densidade da imagem observada dado o campo de rótulos. O método de segmentação busca, portanto, encontrar uma estimativa $\mathbf{x}$ do campo de rótulos utilizando o método MPM, que tem como critério a minimização do número de pixels rotulados erroneamente. A distribuição de probabilidade da imagem observada $f_{\mathbf{Y}}$ é desconsiderada, já que esta não interfere na otimização de $p_{\mathbf{X},\mathbf{Y}}$. Tratando-se de uma segmentação não-supervisionada, o conjunto de parâmetros $\boldsymbol{\theta}$ é também desconhecido. Utiliza-se então o algoritmo EM para estimá-los por máxima verossimilhança.

¹As notações p e f são utilizadas para descrever distribuições de probabilidade de funções discretas e contínuas, respectivamente.

Os modelos de $p_{\mathbf{X}}$ e $f_{\mathbf{Y}|\mathbf{X}}$, necessários à formulação da distribuição de probabilidade *a posteriori*, serão descritos nas seções 2.1.1 e 2.1.2, respectivamente. Os algoritmos MPM e EM serão apresentados com detalhes nas seções 2.1.3 e 2.1.4, sendo que a combinação destes métodos será mostrada na seção 2.1.5.

2.1.1 Modelo para o Campo de Rótulos

O campo de rótulos $\mathbf{X}$ é modelado como um CAM. Para melhor explicar este modelo, é necessário, primeiramente, definir o conceito de **vizinhança**. Uma coleção $\mathcal{G} = \{\mathcal{G}_s \subseteq S, s \in S\}$ é um sistema de vizinhança para o látice S se, para qualquer pixel $s \in S$, $s \notin \mathcal{G}_s$ e $s \in \mathcal{G}_r \iff r \in \mathcal{G}_s, \forall r \in S$. Os elementos de $\mathcal{G}_s$ são, desta forma, os vizinhos do pixel s . Nas figuras 2.1 (a) e (c) são apresentados dois exemplos de possíveis sistemas de vizinhança.

Um CAM, segundo sua definição [Ros97, Dou98], é um campo aleatório no qual a probabilidade de uma variável aleatória do campo assumir um valor qualquer, dados os valores de todas as outras variáveis deste campo, é igual à probabilidade desta mesma variável assumir um valor qualquer dados apenas seus vizinhos. Assim, o campo de rótulos $\mathbf{X}$ será um CAM em relação a um sistema de vizinhança $\mathcal{G}$ se

- (i) $P(\mathbf{X} = \mathbf{x}) > 0, \forall \mathbf{x} \in \Omega_{\mathbf{X}}$ e
- (ii) $P(X_s = x_s | X_r = x_r, r \neq s) = P(X_s = x_s | X_r = x_r, r \in \mathcal{G}_s)$.

Outra informação importante é dada pelo teorema de Hammersley-Clifford [Bes74, GG84], que postula que a probabilidade conjunta de um CAM é $P(\mathbf{X} = \mathbf{x}) = p_{\mathbf{X}}(\mathbf{x})$, onde $p_{\mathbf{X}}$ é uma **distribuição de Gibbs**. Uma distribuição de Gibbs relativa a S e $\mathcal{G}$ é a medida de probabilidade $p_{\mathbf{X}}$ definida em $\Omega_{\mathbf{X}}$ pela expressão

$$p_{\mathbf{X}}(\mathbf{x}) = \frac{1}{Z} \exp \left\{ \frac{-U(\mathbf{x})}{T} \right\}, \quad (2.2)$$

onde

$$Z = \sum_{\mathbf{x} \in \Omega_{\mathbf{X}}} \exp \left\{ \frac{-U(\mathbf{x})}{T} \right\}$$

é o fator de normalização, T representa a **temperatura** e $U(\mathbf{x})$ é chamada de **função de energia** com a forma

$$U(\mathbf{x}) = \sum_{c \in C} V_c(\mathbf{x}). \quad (2.3)$$

$V_c(\mathbf{x})$ é uma função que depende das **cliques**, sendo a família $\{V_c : c \in C\}$ chamada de **potencial**. No campo de rótulos, o conjunto $C \subseteq S$ de pixels é uma clique se, para quaisquer pixels $r, s \in C$, $s \in \mathcal{G}_r$, ou seja, todos são vizinhos entre si. Nota-se que a coleção de cliques C é induzida pelo sistema de vizinhança $\mathcal{G}$. Nas figuras 2.1 (b) e (d) é possível visualizar as cliques de um pixel na vizinhança 4 e 8-conectada, respectivamente.

Uma função energia comumente empregada em estudos de segmentação de imagens [LD89, Pap92, CD00] pode ser definida como

$$U(\mathbf{x}) = \sum_{\{r,s\} \in C} \beta t(x_r, x_s) + \sum_{\{r\} \in C} \gamma_{x_r}, \quad (2.4)$$

Figura 2.1: (a) Vizinhança 4-conectada e (b) suas respectivas cliques. (c) Vizinhança 8-conectada e (d) suas respectivas cliques.

onde

$$t(x_r, x_s) = \begin{cases} 0, & \text{se } x_r = x_s, \\ 1, & \text{se } x_r \neq x_s. \end{cases} \quad (2.5)$$

Neste modelo é utilizada a vizinhança 4-conectada. Nota-se que apenas as cliques duplas (dois pixels) e únicas (um pixel) são levadas em consideração. O parâmetro β , chamado de **parâmetro de interação espacial**, define o peso da função de penalidade $t(x_r, x_s)$ de uma clique dupla. Esta função tem por objetivo desestimular pixels vizinhos a terem rótulos diferentes. Os parâmetros $\{\gamma_k\}_{k=1, \dots, L}$, que influem nas cliques únicas, podem ser considerados como os custos de ocorrência de cada rótulo k .

Dada as equações (2.2) e (2.4) e utilizando temperatura constante $T = 1$, a probabilidade *a priori* do campo de rótulos pode ser escrita como

$$p_{\mathbf{x}}(\mathbf{x}) = \frac{1}{Z} \exp \left(- \sum_{\{r,s\} \in C} \beta t(x_r, x_s) - \sum_{\{r\} \in C} \gamma_{x_r} \right). \quad (2.6)$$

Outro resultado importante que deve ser formulado é a probabilidade condicional $P(X_s = x_s | X_r = x_r, r \in \mathcal{G}_s)$. Segundo a definição, dados dois eventos A e B em um espaço de probabilidade, a probabilidade condicional de A ocorrer dado que B ocorreu é

$$P(A|B) = \frac{P(A \cap B)}{P(B)}. \quad (2.7)$$

Utilizando a equação (2.7) e cancelando termos no numerador e denominador, tem-se

$$P(X_s = x_s | X_r = x_r, r \in \mathcal{G}_s) = \frac{1}{Z_s} \exp \left(- \frac{1}{T} \sum_{c:s \in C} V_c(\mathbf{x}^{(k)}) \right), \quad (2.8)$$

onde

$$Z_s = \sum_{k=1}^L \exp \left(- \frac{1}{T} \sum_{c:s \in C} V_c(\mathbf{x}^{(k)}) \right)$$

e $\mathbf{x}^{(k)}$ é a configuração em que $x_s = k$ e que coincide com $\mathbf{x}$ nas demais posições. Vale ressaltar que o lado direito da expressão depende apenas de $x_s = k$ e $x_r, r \in \mathcal{G}_s$.

2.1.2 Modelo para a Imagem Observada

Analisando a equação (2.1), nota-se a necessidade de se obter a função $f_{\mathbf{Y}|\mathbf{X}}$, que representa o modelo da imagem observada $\mathbf{Y}$ condicionado no campo de rótulos. Assume-se inicialmente que as variáveis aleatórias $Y_1, \dots, Y_N$ são condicionalmente independentes dado o campo de rótulos $\mathbf{X}$. Assume-se também que a função densidade de probabilidade condicional de Y_s dado $\mathbf{X}$ depende apenas do valor de X_s , ou seja, a intensidade do nível de cinza de um pixel depende apenas dos parâmetros do rótulo ao qual o mesmo pertence. Desta forma, a função $f_{\mathbf{Y}|\mathbf{X}}$ pode ser escrita da seguinte maneira:

$$f_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}, \theta) = \prod_{s=1}^N f_{Y_s|\mathbf{X}, \theta}(y_s|\mathbf{x}, \theta) = \prod_{s=1}^N f_{Y_s|X_s, \theta}(y_s|x_s, \theta). \quad (2.9)$$

Assim como em outros trabalhos sobre segmentação propostos anteriormente [Pap92, ZML94], a função $f_{Y_s|X_s}$ assume a forma de uma distribuição normal (gaussiana). Desta forma, para cada rótulo $k = 1, \dots, L$, existem dois parâmetros: média (μ_k) e variância (σ_k^2). O vetor de parâmetros θ possui, portanto, a média e a variância de cada rótulo, ou seja, $\theta = [\mu_1, \sigma_1^2, \dots, \mu_L, \sigma_L^2]$. Utilizando a equação (2.9), a função densidade de probabilidade condicional de $\mathbf{Y}$ dado $\mathbf{X}$ pode ser escrita como

$$f_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}, \theta) = \prod_{s=1}^N \frac{1}{\sqrt{2\pi\sigma_{x_s}^2}} \exp\left(-\frac{(y_s - \mu_{x_s})^2}{2\sigma_{x_s}^2}\right). \quad (2.10)$$

A equação (2.10) interpretada como função de θ é chamada de **função de verossimilhança**.

2.1.3 Algoritmo MPM

O algoritmo MPM é uma técnica de relaxação (processo que tenta diminuir a quantidade de energia do sistema) que visa encontrar uma estimativa $\mathbf{x}$ do campo de rótulos por meio de uma otimização estatística baseada na distribuição de probabilidade *a posteriori* $p_{\mathbf{X}|\mathbf{Y}}$. Esta função é obtida substituindo-se as equações (2.6) e (2.10) na equação (2.1). Desta forma, tem-se

$$\begin{aligned} p_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y}, \theta) &= \frac{p_{\mathbf{X}}(\mathbf{x}) \times f_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}, \theta)}{f_{\mathbf{Y}}(\mathbf{y}|\theta)} \\ &= \frac{1}{f_{\mathbf{Y}}(\mathbf{y}|\theta)} \left[\prod_{s=1}^N \frac{1}{\sqrt{2\pi\sigma_{x_s}^2}} \exp\left(-\frac{(y_s - \mu_{x_s})^2}{2\sigma_{x_s}^2}\right) \right] \frac{1}{Z} \exp\left(-\sum_{\{r,s\} \in C} \beta t(x_r, x_s) - \sum_{\{r\} \in C} \gamma_{x_r}\right) \\ &= \frac{1}{Z f_{\mathbf{Y}}(\mathbf{y}|\theta)} \left[\prod_{s=1}^N \frac{1}{\sqrt{2\pi\sigma_{x_s}^2}} \right] \exp\left(-\sum_{s=1}^N \frac{(y_s - \mu_{x_s})^2}{2\sigma_{x_s}^2} - \sum_{\{r,s\} \in C} \beta t(x_r, x_s) - \sum_{\{r\} \in C} \gamma_{x_r}\right) \end{aligned}$$

$$\begin{aligned}
&= \frac{1}{Z f_{\mathbf{Y}}(\mathbf{y}|\boldsymbol{\theta})} \exp \left(-\frac{1}{2} \sum_{s=1}^N \log(2\pi\sigma_{x_s}^2) - \sum_{s=1}^N \frac{(y_s - \mu_{x_s})^2}{2\sigma_{x_s}^2} - \sum_{\{r,s\} \in C} \beta t(x_r, x_s) - \sum_{\{r\} \in C} \gamma_{x_r} \right) \\
&= \frac{1}{Z f_{\mathbf{Y}}(\mathbf{y}|\boldsymbol{\theta})} \exp \left(- \sum_{\{r\} \in C} \left[\frac{1}{2} \log(2\pi\sigma_{x_r}^2) + \frac{(y_r - \mu_{x_r})^2}{2\sigma_{x_r}^2} \right] - \sum_{\{r,s\} \in C} \beta t(x_r, x_s) - \sum_{\{r\} \in C} \gamma_{x_r} \right). \tag{2.11}
\end{aligned}$$

Comparando as equações (2.2) e (2.11), nota-se que $p_{\mathbf{X}|\mathbf{Y}}$ é também uma distribuição de Gibbs. Assumindo que o vetor $\boldsymbol{\theta}$ é conhecido, o problema de segmentação é formulado como um problema de otimização. O critério utilizado é a minimização do valor esperado do número de pixels erroneamente classificados [MMP87].

A imagem é segmentada minimizando o valor esperado condicional da função de custo $R(\mathbf{X}, \mathbf{x})$, dada a imagem observada $\mathbf{Y}$, sobre todas as possíveis realizações do campo de rótulos $\mathbf{X}$, ou seja, sobre todo $\mathbf{x} \in \Omega_{\mathbf{X}}$. A função custo utilizada é

$$R(\mathbf{X}, \mathbf{x}) = \sum_{s=1}^N t(X_s, x_s), \tag{2.12}$$

onde $t(\cdot, \cdot)$ é a função definida na equação (2.5). Desta forma, $R(\mathbf{X}, \mathbf{x})$ é o número de pixels na qual $\mathbf{X}$ e $\mathbf{x}$ não são iguais, isto é, o número de pixels erroneamente classificados. O valor de $\mathbf{x}$ que minimiza a esperança condicional desta função de custo é denotada como $\mathbf{x}^*$. Assim,

$$E[R(\mathbf{X}, \mathbf{x}^*)|\mathbf{Y} = \mathbf{y}] \leq E[R(\mathbf{X}, \mathbf{x})|\mathbf{Y} = \mathbf{y}] \quad \forall \mathbf{x} \in \Omega_{\mathbf{X}}. \tag{2.13}$$

Utilizando as equações (2.5) e (2.12) obtém-se

$$\begin{aligned}
E[R(\mathbf{X}, \mathbf{x})|\mathbf{Y} = \mathbf{y}] &= E\left[\sum_{s=1}^N t(X_s, x_s)|\mathbf{Y} = \mathbf{y}\right] \\
&= \sum_{s=1}^N E[t(X_s, x_s)|\mathbf{Y} = \mathbf{y}] \\
&= \sum_{s=1}^N P(X_s \neq x_s|\mathbf{Y} = \mathbf{y}) \\
&= \sum_{s=1}^N (1 - P(X_s = x_s|\mathbf{Y} = \mathbf{y})) \\
&= N - \sum_{s=1}^N P(X_s = x_s|\mathbf{Y} = \mathbf{y}). \tag{2.14}
\end{aligned}$$

É possível minimizar este somatório escolhendo para cada posição $s \in S$ o valor de $\mathbf{x}_s$ do conjunto $\{1, \dots, L\}$ que minimize $(1 - P(X_s = x_s|\mathbf{Y} = \mathbf{y}))$ ou, de maneira equivalente, o valor que maximize $P(X_s = x_s|\mathbf{Y} = \mathbf{y})$. A estimativa do campo de rótulos $\mathbf{X}$, gerada pela maximização independente desta função densidade de probabilidade condicional marginal para cada $s \in S$, é conhecida como **maximização das marginais a posteriori** [MMP87], sendo denotada $\mathbf{x}_{MPM}$. As equações (2.13) e (2.14) mostram que

$\mathbf{x}_{MPM} = \mathbf{x}^*$. Entretanto, $\mathbf{x}_{MPM}$ minimiza sobre todo $\mathbf{x}$ o número esperado de pixels erroneamente classificados. Para estimar $\mathbf{x}_{MPM}$, é necessário encontrar para cada $s \in S$ o valor do rótulo k que maximize

$$P(X_s = k | \mathbf{Y} = \mathbf{y}, \boldsymbol{\theta}) = p_{X_s | \mathbf{Y}}(k | \mathbf{y}, \boldsymbol{\theta}) = \sum_{\mathbf{x} \in \Omega_{k,s}} p_{\mathbf{X} | \mathbf{Y}}(\mathbf{x} | \mathbf{y}, \boldsymbol{\theta}), \quad (2.15)$$

onde $k \in \{1, \dots, L\}$ e $\Omega_{k,s} = \{\mathbf{x} \in \Omega_{\mathbf{x}} : x_s = k\}$. Nota-se que equação (2.15) nada mais é do que a distribuição marginal condicional (*a posteriori*) do rótulo k do pixel s .

Visto que a cardinalidade de $\Omega_{k,s}$ é L^{N-1} , isto é, existem L^{N-1} possibilidades de configurações de campos de rótulos diferentes, o cálculo exato destas probabilidades é computacionalmente inviável. O método MPM procura, portanto, aproximá-las da seguinte maneira:

- Utilizar o amostrador de Gibbs (*Gibbs sampler*) para gerar uma cadeia de Markov $\mathbf{X}(t)$ que convirja em distribuição para um CAM com probabilidade $p_{\mathbf{X} | \mathbf{Y}}$;
- Aproximar $P(X_s = k | \mathbf{Y} = \mathbf{y}, \boldsymbol{\theta})$ como sendo a fração de tempo (frequência relativa) em que o pixel s da cadeia $\mathbf{X}(t)$ passa no rótulo k , para quaisquer k e s .

A cada passo do amostrador de Gibbs apenas um pixel é visitado. Logo, $\mathbf{X}(t-1)$ e $\mathbf{X}(t)$ podem diferir no máximo em uma posição. No tempo t , o estado de $\mathbf{X}(t)$ no pixel s é uma variável aleatória $X_s(t) \in \{1, \dots, L\}$. Seja $q_t \in S$ o pixel visitado no instante t . O estado de $X_{q_t}(t)$ é amostrado da distribuição $P(X_{q_t} = k | \mathbf{Y} = \mathbf{y}, X_r = x_r(t-1), r \in \mathcal{G}_{q_t}, \boldsymbol{\theta})$, que pode ser calculada utilizando as equações (2.8), (2.10) e (2.11). Com efeito,

$$\begin{aligned} P(X_{q_t} = k | \mathbf{Y} = \mathbf{y}, X_r = x_r(t-1), r \in \mathcal{G}_{q_t}, \boldsymbol{\theta}) \\ = \frac{1}{Z^*} \exp \left(-\frac{1}{2} \log(2\pi\sigma_k^2) - \frac{(y_{q_t} - \mu_k)^2}{2\sigma_k^2} - \sum_{\{q_t, r\} \in C} \beta t(k, x_r(t-1)) - \sum_{\{r\} \in C} \gamma_{x_r(t-1)} \right) \\ = \frac{1}{Z^*} \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp \left(-\frac{(y_{q_t} - \mu_k)^2}{2\sigma_k^2} - \sum_{\{q_t, r\} \in C} \beta t(k, x_r(t-1)) - \sum_{\{r\} \in C} \gamma_{x_r(t-1)} \right), \end{aligned} \quad (2.16)$$

onde

$$Z^* = \sum_{k=1}^L \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp \left(-\frac{(y_{q_t} - \mu_k)^2}{2\sigma_k^2} - \sum_{\{q_t, r\} \in C} \beta t(k, x_r(t-1)) - \sum_{\{r\} \in C} \gamma_{x_r(t-1)} \right)$$

e

$$t(k, x_r(t-1)) = \begin{cases} 0, & \text{se } x_r(t-1) = k, \\ 1, & \text{se } x_r(t-1) \neq k. \end{cases}$$

Se a sequência $\{q_1, q_2, \dots\}$ contém cada pixel $s \in S$ infinitas vezes, então para qualquer configuração inicial $\mathbf{x}(0) \in \Omega_{\mathbf{X}}$,

$$\lim_{t \rightarrow \infty} P(\mathbf{X}(t) | \mathbf{Y} = \mathbf{y}, \mathbf{X}(0) = \mathbf{x}(0)) = p_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y}, \theta), \quad \forall \mathbf{x} \in \Omega_{\mathbf{X}}. \quad (2.17)$$

A aproximação da probabilidade marginal condicional em cada pixel requer a seguinte definição:

$$u_{k,s}(t) = \begin{cases} 1, & \text{se } X_s(t) = k, \\ 0, & \text{se } X_s(t) \neq k. \end{cases} \quad (2.18)$$

Então, sendo T_s o número de visitas do amostrador de Gibbs ao pixel s , é possível calcular

$$p_{X_s|\mathbf{Y}}(k|\mathbf{y}, \theta) = P(X_s = k | \mathbf{Y} = \mathbf{y}, \theta) \approx \frac{1}{T_s} \sum_{t=1}^{T_s} u_{k,s}(t) \quad (2.19)$$

para quaisquer pixel s e rótulo k . Vale notar que o somatório nada mais é do que a contagem do número de vezes em que foi atribuído o rótulo k ao pixel s .

Amostrador de Gibbs

A técnica do amostrador de Gibbs tem como objetivo a geração indireta das variáveis aleatórias de uma determinada distribuição conjunta, sem a necessidade efetiva de calculá-las [CG92]. Este método utiliza as distribuições condicionais para o cálculo indireto de uma distribuição conjunta complexa.

Para explicar o funcionamento desta técnica, considera-se primeiramente um modelo com duas variáveis aleatórias A e B e a respectiva densidade conjunta $f(a, b)$. Sendo conhecidas as distribuições condicionais $f(A = a | B = b)$ e $f(B = b | A = a)$, geram-se amostras alternadas destas funções. A amostra de uma variável deve ser condicionada aos valores obtidos das variáveis na iteração anterior, caracterizando uma cadeia de Markov. Desta forma, para este modelo, a distribuição conjunta pode ser calculada pelo limite da cadeia

$$B_0 = b_0, A_0 = a_0, B_1 = b_1, A_1 = a_1, B_2 = b_2, A_2 = a_2, \dots, B_K = b_K, A_K = a_K.$$

O valor inicial da cadeia B_0 deve ser definido como um valor aleatório enquanto que os valores restantes são obtidos iterativamente da seguinte maneira:

$$\begin{aligned} A_i &\sim f(a | B_i = b_i), \\ B_i &\sim f(b | A_{i-1} = a_{i-1}). \end{aligned} \quad (2.20)$$

A geração desta cadeia é denominada **amostragem de Gibbs**. À medida que cresce o número de iterações ($K \rightarrow \infty$), a distribuição associada a A_K converge para $f(a)$. Desta forma, é possível determinar as características desejáveis desta marginal.

Esta técnica é utilizada no algoritmo MPM para a geração da cadeia $\mathbf{X}(t)$. Em cada passo t , gera-se o valor de apenas uma variável aleatória X_{q_t} . Este valor é obtido retirando-se uma amostra da probabilidade condicional $P(X_{q_t} = k | \mathbf{Y} = \mathbf{y}, X_r = x_r(t-1), r \in \mathcal{G}_{q_t}, \theta)$ dada na equação (2.16), utilizando-se apenas os valores das variáveis vizinhas da cadeia $\mathbf{X}(t-1)$, gerada na iteração anterior. Desta forma, a única diferença entre $\mathbf{X}(t)$ e $\mathbf{X}(t-1)$ está na variável X_{q_t} . Este processo pode ser assim resumido:

1. Escolher um pixel X_{q_t} .
2. Classificar esse pixel no mapa de rótulos com o valor obtido na geração de uma amostra com probabilidade $P(X_{q_t} = k | Y = y, X_r = x_r(t-1), r \in \mathcal{G}_{q_t}, \boldsymbol{\theta})$;
3. Repetir os passos 1 e 2 para um número determinado de iterações para que cada pixel seja visitado pelo menos duas vezes.

2.1.4 Algoritmo EM

O algoritmo EM [DLR77, Bil98] é um método de estimação de parâmetros que se enquadra no esquema de estimação por **máxima verossimilhança**. Ele busca maximizar o valor da função de verossimilhança em problemas de dados incompletos. Como esta função é baseada em um produtório, a maximização pode ser feita no seu logaritmo, pois esta operação a simplifica para um somatório de logaritmos, sem alterar seu ponto de máximo.

Dentro do contexto de segmentação de imagens, o objetivo do método EM é estimar iterativamente o vetor $\boldsymbol{\theta}$ de parâmetros. Em cada iteração são realizados dois passos: (E) o cálculo da esperança e (M) sua maximização. Assim, sendo a imagem $\mathbf{Y}$ o dado observado, o campo de rótulos $\mathbf{X}$ o dado faltante e $\boldsymbol{\theta}(p)$ a estimativa de $\boldsymbol{\theta}$ na p -ésima iteração do processo, deve-se calcular

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}(p-1)) = E[\log f_{\mathbf{X}, \mathbf{Y}}(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta}) | \mathbf{Y} = \mathbf{y}, \boldsymbol{\theta}(p-1)] \quad (2.21)$$

para encontrar o valor de $\boldsymbol{\theta}$ que maximize $Q(\boldsymbol{\theta}(p), \boldsymbol{\theta}(p-1))$, ou seja,

$$\boldsymbol{\theta}(p) = \arg \max_{\boldsymbol{\theta} \in \Omega_{\boldsymbol{\theta}}} Q(\boldsymbol{\theta}(p), \boldsymbol{\theta}(p-1)), \quad (2.22)$$

onde $\Omega_{\boldsymbol{\theta}}$ é o espaço das possíveis realizações de $\boldsymbol{\theta}$. As equações (2.21) e (2.22) representam os passos E e M, respectivamente. Neste problema, tem-se que $f_{\mathbf{X}, \mathbf{Y}}(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta}) = f_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) \times p_{\mathbf{X}}(\mathbf{x})$. Já que a probabilidade $p_{\mathbf{X}}$ não depende de $\boldsymbol{\theta}$, basta considerar $f_{\mathbf{Y}|\mathbf{X}}$ dada na equação (2.10). Assim,

$$\log f_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) = K - \frac{1}{2} \sum_{s=1}^N \log \sigma_{x_s}^2 - \frac{1}{2} \sum_{s=1}^N \frac{(y_s - \mu_{x_s})^2}{\sigma_{x_s}^2}, \quad (2.23)$$

onde K é uma constante que não interfere na maximização e, por isso, pode ser descartada. Assim, a esperança condicional em $\mathbf{Y}$ da equação (2.23) deve ser calculada. Considerando a distribuição marginal condicional $P(X_s = k | \mathbf{Y} = \mathbf{y}, \boldsymbol{\theta})$, $k = 1, \dots, L$, tem-se

$$Q(\boldsymbol{\theta}, \boldsymbol{\theta}(p-1)) = -\frac{1}{2} \sum_{s=1}^N \left[\sum_{k=1}^L \left(\log \sigma_k^2 + \frac{(y_s - \mu_k)^2}{\sigma_k^2} \right) P(X_s = k | \mathbf{Y} = \mathbf{y}, \boldsymbol{\theta}(p-1)) \right]. \quad (2.24)$$

Diferenciando, igualando a zero e resolvendo para $\boldsymbol{\theta}(p) = [\mu_1, \sigma_1^2, \dots, \mu_L, \sigma_L^2]$, obtém-se

$$\mu_k(p) = \frac{1}{N_k(p)} \sum_{s=1}^N y_s P(X_s = k | \mathbf{Y} = \mathbf{y}, \boldsymbol{\theta}(p-1)) \quad (2.25)$$

e

$$\sigma_k^2(p) = \frac{1}{N_k(p)} \sum_{s=1}^N (y_s - \mu_k(p))^2 P(X_s = k | \mathbf{Y} = \mathbf{y}, \boldsymbol{\theta}(p-1)), \quad (2.26)$$

onde

$$N_k(p) = \sum_{s=1}^N P(X_s = k | \mathbf{Y} = \mathbf{y}, \boldsymbol{\theta}(p-1)). \quad (2.27)$$

A cada iteração, o algoritmo deve passar pelas equações (2.25) a (2.27), encontrando, assim, estimativas de média e variância de cada rótulo, representadas pelo vetor $\boldsymbol{\theta}$ de parâmetros. Como visto nessas equações, para executar esse cálculo é necessário estimar, primeiramente, as probabilidades $P_{X_s|\mathbf{Y}}$. Estes valores são obtidos do algoritmo MPM na busca da estimativa $\mathbf{x}_{MPM}$. Nota-se o algoritmo EM é monotônico, já que a cada iteração a função de máxima verossimilhança não diminui.

2.1.5 Algoritmo EM/MPM

Tendo definido o número de rótulos L , os valores β , γ_k , $k = 1, \dots, L$ e o valor inicial do vetor de parâmetros $\boldsymbol{\theta}$, o algoritmo EM/MPM combina as técnicas descritas nas seções 2.1.3 e 2.1.4 para encontrar um campo de rótulos $\mathbf{X}$ que representa a segmentação da imagem $\mathbf{Y}$. Primeiramente, executa-se o algoritmo MPM com a estimativa inicial $\hat{\boldsymbol{\theta}}(0)$. Desta forma, as primeiras estimativas $p_{X_s|\mathbf{Y}}(k|\mathbf{y}, \hat{\boldsymbol{\theta}}(0))$ das probabilidades de cada pixel podem ser utilizadas nas equações (2.25) a (2.27) para encontrar uma versão atualizada de $\boldsymbol{\theta}$, ou $\hat{\boldsymbol{\theta}}(1)$. Novamente, executa-se o algoritmo MPM para encontrar as estimativas $p_{X_s|\mathbf{Y}}(k|\mathbf{y}, \hat{\boldsymbol{\theta}}(1))$ que, por sua vez, são utilizadas para a estimação EM de $\hat{\boldsymbol{\theta}}(2)$. Este processo é continuado por um número $\mathcal{P}$ de iterações.

A cada iteração do algoritmo MPM é gerada uma cadeia de Markov $\mathbf{X}(t)$ que converge, em distribuição, para um campo aleatório com probabilidade $p_{\mathbf{X}|\mathbf{Y}}$. Os elementos desta cadeia são os campos $\mathbf{X}(0), \dots, \mathbf{X}(t), \dots, \mathbf{X}(\mathcal{T} \times N)$ que se diferem por apenas um pixel, onde $\mathcal{T}$ é o número de vezes em que cada pixel foi visitado pelo amostrador de Gibbs. Ao longo do processo EM/MPM são geradas várias destas cadeias, formando o conjunto $\mathbf{X}(1, t), \dots, \mathbf{X}(\mathcal{P} + 1, t)$, para algum $\mathcal{P} \geq 1$. A geração da cadeia $\mathbf{X}(p, t)$, correspondente à iteração p do processo e à estimativa do vetor $\boldsymbol{\theta}$, obtida durante o estágio p , é representada pela variável aleatória $\boldsymbol{\Theta}(p)$. O algoritmo inicia com uma estimativa $\boldsymbol{\Theta}(0) = \hat{\boldsymbol{\theta}}(0)$ para algum $\hat{\boldsymbol{\theta}}(0) \in \Omega_{\boldsymbol{\theta}}$. A primeira cadeia de Markov $\mathbf{X}(1, t)$ é gerada utilizando o procedimento descrito na seção 2.1.3. O estado de $X_{q_t}(1, t)$ é determinado pela amostragem de um valor da distribuição condicional $P(X_{q_t} = k | \mathbf{Y} = \mathbf{y}, X_r = x_r(1, t-1), r \in \mathcal{G}_{q_t}, \hat{\boldsymbol{\theta}}(0))$ como dado na equação (2.16).

Após cada pixel ter sido visitado $\mathcal{T}$ vezes durante o estágio p do processo, para algum $\mathcal{T} > 1$, o valor aproximado da densidade marginal $P(X_s = k | \mathbf{Y} = \mathbf{y}, \boldsymbol{\Theta} = \hat{\boldsymbol{\theta}}(0))$ é calculado utilizando as equações (2.18) e (2.19):

$$P(X_s = k | \mathbf{Y} = \mathbf{y}, \boldsymbol{\Theta} = \hat{\boldsymbol{\theta}}(0)) \approx v_{k,s}(1, t) = \frac{1}{t} \sum_{i=1}^t u_{k,s}(1, i), \quad (2.28)$$

onde

$$u_{k,s}(1, i) = \begin{cases} 1, & \text{se } X_s(1, i) = k, \\ 0, & \text{se } X_s(1, i) \neq k. \end{cases}$$

Assim, o vetor de parâmetros $\Theta(1) = [M_1(1), S_1(1), \dots, M_L(1), S_L(1)]$, onde $M_k(1)$ e $S_k(1)$ são a média e a variância do rótulo k na primeira iteração, é obtido por meio das equações (2.25) a (2.27), calculando-se

$$M_k(1) = \frac{\sum_{s=1}^N y_s \times v_{k,s}(1, t)}{\sum_{s=1}^N v_{k,s}(1, t)} \quad (2.29)$$

e

$$S_k(1) = \frac{\sum_{s=1}^N (y_s - M_k(1))^2 \times v_{k,s}(1, t)}{\sum_{s=1}^N v_{k,s}(1, t)}. \quad (2.30)$$

Calculada a estimativa $\hat{\theta}(1)$, o processo continua com a geração da cadeia $\mathbf{X}(p, t)$, seguida da estimação de $\Theta(p) = \hat{\theta}(p)$, até que um número $\mathcal{P}$ de iterações seja atingido. Desta maneira, os passos do algoritmo podem ser assim resumidos:

1. Escolher valores iniciais para $\Theta(0)$ e $\mathbf{X}(1, 0)$. Fazer $p = 1$;
2. Executar o algoritmo MPM utilizando $\hat{\theta}(p - 1)$ e visitando $\mathcal{T}$ vezes cada pixel;
3. Utilizar as equações (2.29) e (2.30) do algoritmo EM para obter $\hat{\theta}(p)$;
4. Repetir os passos 2 e 3 para $\mathcal{P}$ iterações.

A segmentação final é obtida executando-se o algoritmo MPM pela última vez com a estimação de θ obtida no estágio $\mathcal{P}$, escrita como $\hat{\theta}(\mathcal{P})$. O último elemento $\mathbf{X}(\mathcal{P} + 1, \mathcal{T} \times N)$ da última cadeia de Markov $\mathbf{X}(\mathcal{P} + 1, t)$ produzida é tido como o campo de rótulos final, ou seja, a imagem segmentada. A escolha de $\mathbf{X}(1, 0)$ é feita de maneira aleatória e independente em cada posição $s \in S$. Já a escolha de $\Theta(0)$ pode ser feita escolhendo-se, para cada rótulo, um valor fixo qualquer para as variâncias $\sigma_k^2(0)$ e o valor das médias $\mu_k(0)$, distribuídas entre 0 e 255 da seguinte maneira:

$$\mu_k(0) = \frac{128}{L} + \frac{255 \times (k - 1)}{L}, \quad k = 1, \dots, L. \quad (2.31)$$

A estimativa inicial dessas médias pode alterar o resultado da segmentação. Uma alternativa para o cálculo de $\mu_k(0)$, proposta neste trabalho, pode ser vista na seção 3.2.

2.2 Segmentação EM/MPM Multiresolução

Uma cena, visualizada em curta distância, permite que o observador perceba uma grande quantidade de detalhes. Entretanto, em determinadas situações, essa visão deta-

Figura 2.2: Imagem com combinação de duas texturas mostrada em diversas resoluções. A cada escala menor as texturas se tornam mais suaves.

Figura 2.3: Látice multiresolução com três níveis, 1 (base), 2 (intermediário) e 3 (topo).

lhada pode não ser necessária, ou pior, pode gerar confusão no seu entendimento global. Nestes casos, uma boa alternativa é distanciar-se da cena para observar o todo.

Na área de processamento de imagens, este processo é chamado de segmentação por multiresolução ou multiescala. De maneira análoga, muitos estudos de segmentação por textura [CD99, BS94] fazem uso de escalas mais baixas da imagem a fim de obter características mais gerais e homogêneas dos objetos. Diminuindo a resolução, uma macrotextura se transforma em uma microtextura, facilitando sua análise, como visto no exemplo da figura 2.2. Além disso, esse artifício pode diminuir o tempo de processamento do processo, uma vez que a maior parte do trabalho é realizado em resoluções pequenas, onde a quantidade de pixels é consideravelmente menor.

Alguns modelos, propostos por Mary L. Comer e Eduard J. Delp, foram desenvolvidos para a formulação do algoritmo EM/MPM multiresolução [CD95, Com95, CD99]. Nessa abordagem, a segmentação da imagem é feita em um **látice multiresolução** S com M níveis, semelhante a uma pirâmide, representado na figura 2.3. A base desse látice (nível $n = 1$), que contém a resolução mais fina (com maior detalhamento), é a imagem original. Nos níveis acima ($n = 2, \dots, M$) estão as resoluções mais grossas, onde cada pixel corresponde a quatro pixels do nível abaixo.

Como já foi dito, a imagem original é a base da pirâmide. O valor de cada pixel do nível imediatamente acima é obtido fazendo a média de quatro pixels do nível abaixo. Este processo é continuado para um número M de níveis, previamente definido. Esse

novo esquema induz a uma nova modelagem do problema, apresentada na seção 2.2.1. O algoritmo de segmentação, apesar de semelhante ao método sem multiresolução, sofre algumas modificações, como apresentado na seção 2.2.2.

2.2.1 Modelagem Multiresolução

O modelo utilizado para o campo de rótulos na abordagem multiresolução é uma extensão do modelo de uma resolução descrito na seção 2.1.1. Desta maneira, $\mathbf{X}$ é um CAM com função de probabilidade dada pela equação (2.2). A coleção C de cliques definida em S representa agora, além das cliques únicas e duplas do sistema de vizinhança do mesmo látice (tipos 0 e 1, respectivamente), a clique dupla que relaciona o pixel com seu pai (tipo 2), quando este existir. Desta forma, na abordagem multiresolução a função de probabilidade $p_{\mathbf{X}}$ pode ser escrita como

$$p_{\mathbf{X}}(\mathbf{x}) = \frac{1}{Z} \exp \left(- \sum_{\{r,s\} \in C_1} \beta^{(n_r)} t(x_r, x_s) - \sum_{\{r,s\} \in C_2} \alpha^{(n_r)} t(x_r, x_s) - \sum_{\{r\} \in C_0} \gamma_{x_r} \right), \quad (2.32)$$

onde C_0 é o conjunto de cliques únicas, C_1 e C_2 são as coleções de cliques dos tipos 1 e 2, $\beta^{(n_r)}$ e $\alpha^{(n_r)}$ são os parâmetros de interação espacial no nível n do pixel r e a função $t(x_r, x_s)$ é definida como na equação (2.5).

Para a modelagem da imagem observada, que busca obter a função densidade de probabilidade $f_{\mathbf{Y}|\mathbf{X}}$, existem diversas abordagens. É possível, por exemplo, manter o mesmo modelo utilizado no método EM/MPM padrão, ignorando as diversas resoluções. Uma alternativa, estudada por Comer e Delp [CD95, Com95, CD99], consiste em utilizar um modelo autoregressivo (AR) adaptado dos modelos AR temporais, na qual as informações de níveis de cinza são passadas de cima para baixo na pirâmide. Neste trabalho, foi utilizada uma formulação dada por

$$f_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x}, \boldsymbol{\theta}) = \prod_{s=1}^N \frac{1}{\sqrt{2\pi\sigma_{x_s}^{(n_s)^2}}} \exp \left(- \frac{(y_s - \mu_{x_s}^{(n_s)})^2}{2\sigma_{x_s}^{(n_s)^2}} \right), \quad (2.33)$$

onde $\mu_k^{(n)}$ e $\sigma_k^{(n)}$ são, respectivamente, as médias e variâncias do rótulo $k = 1, \dots, L$ no nível $n = 1, \dots, M$. Neste contexto, portanto, o vetor $\boldsymbol{\theta}$ de parâmetros é composto pelas médias e variâncias de cada rótulo para cada nível. Assim,

$$\boldsymbol{\theta} = \left(\mu_1^{(1)}, \sigma_1^{(1)^2}, \dots, \mu_L^{(1)}, \sigma_L^{(1)^2}, \dots, \mu_1^{(M)}, \sigma_1^{(M)^2}, \dots, \mu_L^{(M)}, \sigma_L^{(M)^2} \right). \quad (2.34)$$

2.2.2 Algoritmo EM/MPM Multiresolução

Existem algumas possíveis abordagens para o algoritmo EM/MPM multiresolução. Uma delas consiste em segmentar um nível por vez, começando pelo mais alto. As informações do mapa de rótulos obtidas nesta etapa são utilizadas na segmentação do nível abaixo e assim por diante. Executa-se portanto, um número $\mathcal{P}^{(M)}$ de iterações do algoritmo para a resolução mais grossa para, em seguida, executar $\mathcal{P}^{(M-1)}$ iterações do próximo látice e repetir este processo em todos os níveis.

Uma alternativa, utilizada neste trabalho, é executar a segmentação de todos os pixels da pirâmide em cada iteração do processo EM/MPM multiresolução. Nesta abordagem são executadas $\mathcal{P}$ iterações de um algoritmo similar ao apresentado na seção 2.1.5. Entretanto, devido às alterações nos modelos de $\mathbf{X}$ e $\mathbf{Y}$, tem-se que

$$p_{\mathbf{X}|\mathbf{Y}}(\mathbf{x}|\mathbf{y}, \boldsymbol{\theta}) \propto \frac{1}{Z} \left[\prod_{s=1}^N \frac{1}{\sqrt{2\pi\sigma_{x_s}^{(n_s)^2}}} \right] \times \exp \left(- \sum_{s=1}^N \frac{(y_s - \mu_{x_s}^{(n_s)})^2}{2\sigma_{x_s}^{(n_s)^2}} - \sum_{\{r,s\} \in C_1} \beta^{(n_r)} t(x_r, x_s) - \sum_{\{r,s\} \in C_2} \alpha^{(n_r)} t(x_r, x_s) - \sum_{\{r\} \in C_0} \gamma_{x_r} \right). \quad (2.35)$$

Com base nesta expressão aplica-se o critério da minimização do número de pixels mal classificados, de maneira similar ao algoritmo MPM apresentado na seção 2.1.3. O vetor $\boldsymbol{\theta}$ de parâmetros deve ser estimado de maneira similar ao algoritmo apresentado na seção 2.1.4. Desta forma, para cada nível $n = 1, \dots, M$, tem-se que

$$\mu_k^{(n)}(p) = \frac{1}{N_k^{(n)}(p)} \sum_{s_n=1}^{N^{(n)}} y_{s_n} P(X_{s_n} = k | \mathbf{Y} = \mathbf{y}, \boldsymbol{\theta}(p-1)) \quad (2.36)$$

e

$$\sigma_k^{(n)^2}(p) = \frac{1}{N_k^{(n)}(p)} \sum_{s_n=1}^{N^{(n)}} (y_{s_n} - \mu_k^{(n)}(p))^2 P(X_{s_n} = k | \mathbf{Y} = \mathbf{y}, \boldsymbol{\theta}(p-1)), \quad (2.37)$$

onde

$$N_k^{(n)}(p) = \sum_{s_n=1}^{N^{(n)}} P(X_{s_n} = k | \mathbf{Y} = \mathbf{y}, \boldsymbol{\theta}(p-1)), \quad (2.38)$$

$N^{(n)}$ é o número de pixels no nível n da pirâmide e s_n é um pixel do nível n .

2.3 Considerações

Neste capítulo foi apresentado o método estocástico não-supervisionado de segmentação de imagens EM/MPM, que combina as técnicas de estimação de parâmetros por máxima verossimilhança (EM) e de segmentação por otimização estatística com o critério da minimização do número pixels erroneamente classificados (MPM). Também foi apresentada uma abordagem multiresolução para o modelo, na qual o sistema de vizinhança se estende aos diversos níveis de escala.

No próximo capítulo será discutida a implementação deste método, bem como as abordagens da estimação inicial das médias e do uso de *annealing*. Já no capítulo 5 será visto a grande variedade de aplicações desta técnica, tal como sua modelagem para a segmentação de malhas de duas e três dimensões, seguida pelos resultados obtidos.

Implementação

Os métodos não-supervisionados de segmentação de imagens descritos no capítulo 2 foram implementados neste trabalho utilizando a linguagem de programação C++ sob plataforma linux, com interface administrada pelo *software* livre Liza [Ron05]. Também foram desenvolvidos e implementados alguns algoritmos e abordagens a fim de melhorar os resultados da segmentação.

Na seção 3.1 deste capítulo serão abordados os detalhes e aspectos da implementação do método EM/MPM. O método proposto para estimação inicial de parâmetros será discutido na seção 3.2. Por fim, o uso da abordagem de *annealing* neste modelo será apresentado na seção 3.3.

3.1 Algoritmo Detalhado

A implementação do método EM/MPM de segmentação de imagens feita nesse trabalho será ilustrada por meio dos pseudo-códigos apresentados nesta seção. No algoritmo 1 é apresentada uma visão geral do processo de segmentação descrito na seção 2.1.5.

Algoritmo 1 EM/MPM

- 1: fazer mapa de rótulos $\mathbf{X}$ aleatório
 - 2: escolher número de rótulos L
 - 3: escolher $\boldsymbol{\theta}(0)$
 - 4: **para cada** iteração $p = 1, \dots, \mathcal{P}$ **faça**
 - 5: $v_{k,s} \leftarrow 0$ para todo pixel s e rótulo k
 - 6: executar o algoritmo MPM com $\boldsymbol{\theta}(p-1)$, obtendo $v_{k,s}$ e $\mathbf{X}(p, t)$
 - 7: executar o algoritmo EM com as estimações $v_{k,s}$, obtendo $\boldsymbol{\theta}(p)$
 - 8: **fim para**
 - 9: a segmentação final é dada pelo último elemento de $\mathbf{X}(p, t)$
-

Na linha 1 deste algoritmo é feita uma escolha uniformemente aleatória dos valores das variáveis do campo de rótulos. Uma outra possível abordagem, não experimentada neste trabalho, é utilizar uma pré-segmentação como estimativa inicial de $\mathbf{X}$. Uma abordagem

para escolha de L , processo descrito na linha 2, será apresentada no capítulo 4. A escolha de $\boldsymbol{\theta}(0)$, feita na linha 3, consiste na estimativa inicial das variâncias σ_k^2 e médias μ_k para todo rótulo $k = 1, \dots, L$. Como explicado na seção 2.1.5, para estimar σ_k^2 basta escolher um valor fixo e igual para todo k . Já para μ_k , um estudo para sua estimação será visto na seção 3.2.

O início do processo EM/MPM se dá na linha 4, que inicia o laço das iterações. O critério de parada é simplesmente o número $\mathcal{P}$ de iterações. Na linha 5 é dada a inicialização do vetor $v_{k,s}$ que, assim como na equação (2.28), é a aproximação de $P(X_s = k | \mathbf{Y} = \mathbf{y}, \boldsymbol{\Theta} = \hat{\boldsymbol{\theta}}(0))$ como a fração de tempo em que o pixel s esteve classificado com rótulo k . Esta informação é obtida na linha 6 do algoritmo MPM, e utilizada na linha 7 do algoritmo EM. A segmentação é considerada o estado final do campo de rótulo $\mathbf{X}$ após as iterações, representado no modelo como o último elemento $\mathbf{X}(p, \mathcal{T} \times N)$ da última cadeia de Markov $\mathbf{X}(p, t)$. O processo MPM pode ser visto com mais detalhes no algoritmo 2.

Algoritmo 2 MPM

```

1: para cada visita  $t = 1, \dots, \mathcal{T}$  faça
2:   para cada pixel  $s = 1, \dots, N$  faça
3:      $Z^* \leftarrow 0$ 
4:     para cada rótulo  $k = 1, \dots, L$  faça
5:        $\text{nvd} \leftarrow$  número de vizinhos de  $s$  diferentes de  $k$ 
6:        $\text{fdp}_k \leftarrow \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{(y_s - \mu_k)^2}{2\sigma_k^2} - \beta \times \text{nvd}\right)$ 
7:        $Z^* \leftarrow Z^* + \text{fdp}_k$ 
8:     fim para
9:      $x_s \leftarrow$  amostra de rótulo  $k$  com distribuição  $\text{fdp}_k$ 
10:     $v_{k,s} \leftarrow v_{k,s} + 1$ 
11:   fim para
12: fim para

```

As linhas 1 e 2 devem assegurar que cada pixel seja visitado um número $\mathcal{T}$ de vezes. A ordem de visita utilizada foi a chamada *raster scan*, que percorre os pixels linha por linha, de cima para baixo, da esquerda para a direita. A cada visita, é calculado o vetor fdp_k da função de probabilidade $P(X_{q_t} = k | \mathbf{Y} = \mathbf{y}, X_r = x_r(t-1), r \in \mathcal{G}_{q_t}, \boldsymbol{\theta})$ dada na equação (2.16). Este cálculo corresponde às linhas 3-8 do algoritmo. Nota-se que nesta implementação foram definidos os valores $\gamma_k = 0$ para todo k , já que esta é uma segmentação não-supervisionada. O esquema de vizinhança utilizado foi o de 8-conectado, mas foram consideradas apenas as cliques duplas. Tendo obtido fdp_k , gera-se uma amostra k com esta distribuição na linha 9. Finalmente, na linha 10, este resultado é acrescentado no vetor $v_{k,s}$.

Ainda neste algoritmo, nota-se que a linha 6 contém o cálculo mais custoso computacionalmente. Nota-se também que este cálculo é dado em função apenas do rótulo k , do nível de cinza y e do número de vizinhos diferentes nvd . Por este motivo, um mesmo valor pode ser calculado muitas vezes durante o processo de maneira ambígua. Este fato induz a formulação do algoritmo 3 que evita o cálculo repetido da fdp_k , otimizando assim o processo MPM.

Desta maneira, antes de iniciar a visita aos pixels, é criado o vetor $\text{tfdp}_{k,y,\text{nvd}}$ calculando $P(X_{q_t} = k | \mathbf{Y} = \mathbf{y}, X_r = x_r(t-1), r \in \mathcal{G}_{q_t}, \boldsymbol{\theta})$ com todas as possíveis combinações de rótulo k , nível de cinza y e número de vizinhos diferentes nvd . Visto que o número de execuções

Algoritmo 3 MPM otimizado

```
1: para cada rótulo  $k = 1, \dots, L$  faça
2:   para cada nível de cinza  $y = 0, \dots, 255$  faça
3:     para cada número de vizinhos diferentes  $\text{nvd} = 0, \dots, 8$  faça
4:        $\text{tfdp}_{k,y,\text{nvd}} \leftarrow \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{(y - \mu_k)^2}{2\sigma_k^2} - \beta \times \text{nvd}\right)$ 
5:     fim para
6:   fim para
7: fim para
8: para cada visita  $t = 1, \dots, T$  faça
9:   para cada pixel  $s = 1, \dots, N$  faça
10:     $Z^* \leftarrow 0$ 
11:    para cada rótulo  $k = 1, \dots, L$  faça
12:       $\text{nvd} \leftarrow$  número de vizinhos de  $s$  diferentes de  $k$ 
13:       $\text{fdp}_k \leftarrow \text{tfdp}_{k,y_s,\text{nvd}}$ 
14:       $Z^* \leftarrow Z^* + \text{fdp}_k$ 
15:    fim para
16:     $x_s \leftarrow$  amostra de rótulo  $K$  com distribuição  $\text{fdp}_k$ 
17:     $v_{K,s} \leftarrow v_{K,s} + 1$ 
18:  fim para
19: fim para
```

dos laços das linhas 8, 9 e 11 ($T \times N \times L$) é, em geral, muito maior que o dos laços das linhas 1, 2 e 3 ($L \times 256 \times 9$), esse algoritmo é, de fato, mais eficiente.

O algoritmo 4 mostra como é feita a geração de uma amostra de rótulo k com distribuição fdp_k . Como as linguagens de programação contam, normalmente, com procedimentos para geração de amostras de maneira exclusivamente uniforme, é necessário utilizar uma função de transferência. Esta função, afdp_k , é a acumulada da fdp_k , que pode receber valores entre 0 e Z^* . Retira-se, então, uma amostra uniforme *random* entre 0 e Z^* e relaciona-se esta ao maior k no qual $\text{afdp}_k \geq \text{random}$.

Algoritmo 4 Amostra de fdp_k

```
1:  $k \leftarrow 1$ 
2:  $\text{afdp} \leftarrow \text{fdp}_1$ 
3:  $\text{random} \leftarrow$  gerar amostra uniforme entre 0 e  $Z^*$ 
4: enquanto  $\text{random} > \text{afdp}$  faça
5:    $k \leftarrow k + 1$ 
6:    $\text{afdp} \leftarrow \text{afdp} + \text{fdp}_k$ 
7: fim enquanto
8:  $x_s \leftarrow k$ 
```

Com os valores $v_{k,s}$ é possível estimar o parâmetro $\theta(p) = [\mu_1, \sigma_1^2, \dots, \mu_L, \sigma_L^2]$ como ilustrado no algoritmo 5. Estes valores são calculados pelas equações (2.29) e (2.30), juntamente com as estimativas de $P(X_s = k | \mathbf{Y} = \mathbf{y}, \theta(p-1))$ para todo $s = 1, \dots, N$ e $k = 1, \dots, L$, representadas pelo vetor $v_{k,s}$.

Algoritmo 5 EM

```
1: para cada rótulo  $k = 1, \dots, L$  faça
2:    $\text{aux}_1 \leftarrow \text{aux}_2 \leftarrow 0$ 
3:   para cada pixel  $s = 1, \dots, N$  faça
4:      $\text{aux}_1 \leftarrow \text{aux}_1 + v_{k,s}$ 
5:      $\text{aux}_2 \leftarrow \text{aux}_2 + v_{k,s} \times y_s$ 
6:   fim para
7:    $\mu_k \leftarrow \frac{\text{aux}_2}{\text{aux}_1}$ 
8:    $\text{aux}_2 \leftarrow 0$ 
9:   para cada pixel  $s = 1, \dots, N$  faça
10:     $\text{aux}_2 \leftarrow (y_s - \mu_k)^2 \times \text{aux}_1$ 
11:  fim para
12:   $\sigma_k^2 \leftarrow \frac{\text{aux}_2}{\text{aux}_1}$ 
13: fim para
```

3.2 Estimação Inicial de Parâmetros

Nos sistemas de análise de imagens não-supervisionados algumas características da imagem devem ser estimadas no início do processo. No algoritmo EM/MPM, como visto na seção 2.1.5, uma informação necessária e que deve ser obtida primeiramente é a estimativa inicial das médias μ_k de cada rótulo k . Esses valores são importantes por interferirem no resultado final da segmentação, como será mostrado no capítulo 5.

A solução para o cálculo desta estimativa proposta por Comer e Delp [CD95, CD00], descrita na equação (2.31), busca manter as médias diferentes entre si e bem distribuídas dentro do domínio dos valores de níveis de cinza. Na figura 3.1 é apresentado, para diferentes valores de L , o valor de μ_k , $k = 1, \dots, L$ calculado por este método.

Figura 3.1: Estimativas iniciais das médias μ_k , $k = 1, \dots, L$ do modelo, para $L = 2, \dots, 8$, utilizando a equação (2.31).

Nota-se que, nesta técnica, os valores iniciais das médias são calculados independentemente da imagem, o que aumenta a chance destes serem muito diferentes das médias

reais. Um exemplo pode ser visto na figura 3.2. Nota-se pelo histograma que existem duas classes (rótulos), representadas pelos dois pontos de máximo. Entretanto, é possível perceber que as estimativas das médias iniciais estão longe dos picos onde mais provavelmente se encontram as médias reais. A média estimada mais à esquerda se confunde com a média real mais à direita. Esta estimativa pode, portanto, interferir negativamente na segmentação final.

Figura 3.2: Histograma de uma imagem e as médias iniciais para $L = 2$ calculadas pela equação (2.31). Nota-se que as estimativas não estão próximas aos picos corretos.

A análise do histograma para estimação inicial dos parâmetros μ_k do modelo apresentado na seção 2.1.2 se mostra plausível. Neste trabalho foi desenvolvido um algoritmo que utiliza limiarização (*thresholding*) para pré-segmentar a imagem e, em seguida, obter uma estimativa inicial das médias. De maneira geral, o método se baseia nos seguintes passos:

1. Obter os valores de limiarização T_j para cada $j = 1, \dots, L - 1$ que dividem o histograma em L diferentes regiões.
2. Definindo $T_0 = 0$ e $T_L = 256$, calcular as médias μ_k para cada $k = 1, \dots, L$ como

$$\mu_k = \frac{\sum_{s|T_{k-1} \leq y_s < T_k} y_s}{\sum_{s|T_{k-1} \leq y_s < T_k} 1}, \quad (3.1)$$

ou seja, a média dos valores do histograma na faixa entre T_{k-1} e T_k .

Uma técnica amplamente utilizada para a determinação automática dos valores de limiarização é a de Otsu [Ots79]. Ela se baseia na escolha dos limiares que maximizam a **variância inter-classes**. Para uma limiarização binária (bi-modal), ou seja, com apenas um limiar separando duas classes, deve-se calcular

$$T = \arg \min_{min < t < max} f(t), \quad (3.2)$$

onde

$$f(t) = w_0(t)w_1(t)[\mu_0(t) - \mu_1(t)]^2,$$

$$w_0(t) = \sum_{i=\min}^t p_i, \quad w_1(t) = 1 - w_0(t), \quad \mu_0(t) = \frac{\sum_{i=\min}^t ip_i}{w_0(t)}, \quad \mu_1(t) = \frac{\sum_{i=\min}^{\max} ip_i - \sum_{i=\min}^t ip_i}{w_1(t)},$$

p_i é a frequência de pixels de intensidade i e $\min$ e $\max$ são os valores mínimo e máximo do nível de cinza, respectivamente. Para uma limiarização multi-modal (com dois ou mais limiares), a formulação mais natural se torna inviável do ponto de vista computacional. Algumas soluções neste contexto podem ser encontradas na literatura [PGPB05, LCC01].

Neste trabalho foi implementado o algoritmo de Otsu recursivo, que utiliza o método bi-modal da equação (3.2) para, em seguida, aplicá-lo novamente nas duas regiões no histograma recentemente separadas, até que o número de regiões seja atingido. Este algoritmo pode ser resumido nos passos a seguir, onde L é o número de regiões pré-determinado, n é o nível da recursão e i é o índice de cada limiar:

1. Fazer $\min = 0$, $\max = 255$, $n = 1$ e $i = 1$;
2. Encontrar o limiar T_i como na equação (3.2);
 - (a) Finalizar caso $L \leq 2^n$;
 - (b) Executar o passo 2 com $\min = \min$, $\max = T_i$ e $i = i + 1$;
 - (c) Executar o passo 2 com $\min = T_i$, $\max = \max$ e $i = i + 1$;
 - (d) Fazer $n = n + 1$.
3. Caso $L < 2^n$, ignorar $2^n - L$ limiares dentre aqueles cujos índices são $2^{n-1} < i < 2^n$, para os quais os valores de $f(T_i)$ são maiores.

A cada execução completa do passo 2, o número de regiões dobra. Assim, quando o número de classes procurada não for potência de dois, é necessário executar o passo 3. Neste, são ignorados alguns dos limiares recém-criados, utilizando como critério o valor da $f(t)$. Desta forma, são preservados os pontos T_i correspondentes aos menores valores da $f(T_i)$. Reordenando os valores T_i para $i = 1, \dots, L - 1$, o pré-processamento é concluído por meio das médias μ_k de acordo com a equação (3.1).

Foi utilizado neste trabalho o histograma com escala logarítmica, que revela com mais detalhes os picos baixos da curva. Isso pode evidenciar classes formadas por poucos pixels que não seriam facilmente identificadas em um histograma tradicional. Outro detalhe foi a utilização de um filtro passa baixa, de média, para eliminar possíveis ruídos no histograma.

3.3 Annealing

O **resfriamento simulado** (*simulated annealing*), originalmente proposto em trabalhos independentes por Kirkpatrick *et al.* [KGV83] e por Cerny [Cer85], é a generalização de um algoritmo de otimização que visa encontrar a orientação de menor energia (mais estável) de um sistema, utilizando para isso o método de Metropolis Monte Carlo [MRR⁺53]. Seu nome é uma analogia com o processo de *annealing* da Física, uma técnica que promove

o aquecimento e um posterior resfriamento controlado de um material com o intuito de moldá-lo e diminuir suas deformações. No aquecimento, os átomos saem de suas posições originais (um mínimo local de energia interna) para que, na fase de resfriamento, tenham mais chances de encontrar posições onde a energia interna é menor.

No método de segmentação EM/MPM apresentado na seção 2.1, o conceito do resfriamento simulado pode ser inserido no contexto da função de probabilidade do campo de rótulos $\mathbf{X}$ dada na equação (2.6). Como dito anteriormente, essa função é uma distribuição de Gibbs, dada na equação (2.2), definida com temperatura constante $T = 1$. Para aprimorar o resultado da segmentação utilizando a técnica do *annealing*, faz-se um aumento dessa temperatura no início do processo, que vai diminuindo gradualmente com as iterações do algoritmo. Alterar a temperatura dessa equação, entretanto, equivale a alterar os parâmetros β e $\gamma_k, k = 1, \dots, L$ da seguinte maneira:

$$p_{\mathbf{X}}(\mathbf{x}) = \frac{1}{Z} \exp \left(- \sum_{\{r,s\} \in C} \beta^* t(x_r, x_s) - \sum_{\{r\} \in C} \gamma_{x_r}^* \right), \quad (3.3)$$

onde

$$\beta^* = \frac{\beta}{T} \text{ e } \gamma_k^* = \frac{\gamma_k}{T}.$$

Desta forma, aumentar a temperatura T significa diminuir os parâmetros β^* e γ_k^* e vice-versa. Fazendo $\beta^* = 0$ e $\gamma_k^* = 0$ para todo $k = 1, \dots, L$, obtém-se a configuração de maior energia desse sistema, causando muita instabilidade no resultado da segmentação. Aumentando-se os valores desses parâmetros, obtém-se resultados mais estáveis.

De fato, utilizar um valor pequeno do parâmetro de interação espacial β^* nas primeiras iterações do algoritmo EM/MPM faz sentido, pois nesse momento o campo de rótulos se encontra bem diferente de uma boa segmentação. Como o primeiro campo de rótulos é aleatório, não existe relação entre o rótulo de um pixel e o de seu vizinho. Assim, o valor da interação entre pixels vizinhos deve ser pequeno. Enquanto o processo é executado, entretanto, o campo vai se aproximando mais do resultado esperado. A interação entre os pixels deve aumentar, já que estes tendem a assumir valores corretos. De maneira análoga, explica-se o aumento contínuo dos parâmetros γ_k^* . No início, os rótulos não estão presentes em seus pesos ideais, mas estes vão se firmando com o desenrolar do processo.

Como dito na seção 3.1, os parâmetros γ_k^* não foram utilizados neste trabalho, sendo atribuídos a eles o valor 0. A implementação do *annealing* foi desenvolvida da seguinte maneira:

1. Definir o valor inicial $\beta^*(0) = \beta_0$, o valor máximo β_{max} e o incremento I ;
2. Executar a iteração p do algoritmo EM/MPM utilizando o valor $\beta^*(p)$;
3. Fazer $\beta^*(p+1) = \beta^*(p) + I$ se $\beta^*(p) < \beta_{max}$;
4. Executar os passos 2 e 3 até que o número de iterações definido seja satisfeito.

Assim, o parâmetro β cresce de forma linear até que um valor máximo seja atingido. A velocidade deste aquecimento se dá em função do valor do incremento escolhido. Esta variação de β pode também ser representada como uma função $\beta^*(p)$ onde

$$\beta^*(p) = \begin{cases} \beta_0 + inc \times p, & \text{se } \beta_0 + inc \times p < \beta_{max}, \\ \beta_{max}, & \text{se } \beta_0 + inc \times p \geq \beta_{max}. \end{cases} \quad (3.4)$$

Validação de Classes

Nos métodos de segmentação não-supervisionados, assim como o EM/MPM, o número de classes (ou **clusters**) distintas presentes na imagem é geralmente assumido como conhecido. Entretanto, em muitas aplicações do mundo real, esse é um parâmetro desconhecido e precisa ser estimado. O problema de se determinar o número L de classes presentes em uma amostragem de dados (neste caso, uma imagem) é conhecido como **validação de classes** (*cluster validation*) e, apesar dos inúmeros estudos a respeito [DH73, GG89], ainda não foi satisfatoriamente resolvido.

Langan, Modestino e Zhang [LMZ98] desenvolveram um estudo acerca de alguns critérios utilizados para a validação de classes em técnicas de segmentação de imagens baseados em modelos estatísticos. Segundo os próprios autores, é possível descrever o número de classes como mais um parâmetro a ser estimado durante a segmentação, embora esta abordagem possa resultar em uma estimação por máxima-verossimilhança de L que se mostra fortemente tendenciosa ao máximo de rótulos considerados.

Uma abordagem alternativa é tratar a validação de classes como um problema de decisão múltipla, empregando critérios sobre as informações resultantes das segmentações realizadas com diferentes candidatos de L , determinando qual L é o melhor. Neste capítulo serão apresentados, na seção 4.1, alguns critérios baseados na função de verossimilhança estudados por Langan *et al.* [LMZ98] e outros, propostos neste trabalho, na seção 4.2.

4.1 Critérios de Validação

Os critérios apresentados nesta seção são baseados na **teoria da informação** (do termo em inglês *information-theoretic criteria*), sendo que a informação que eles fornecem para a validação é derivada de uma métrica extraída do modelo teórico utilizado na classificação. Também são adequados à abordagem dos problemas de decisão múltipla, buscando assim encontrar a ordem do modelo (neste caso, o número de rótulos) que melhor se ajusta aos dados. Esses métodos são relacionados ao princípio da máxima entropia, nos termos da forma geral proposta por Sclove [Sc87], na qual a estimativa da ordem do modelo L é dada por

Figura 4.1: (a) Imagem sintética a ser segmentada utilizando diferentes valores de L . Percebe-se que o número verdadeiro de rótulos é $L = 5$. (b) Imagem de rosto de um macaco babuíno. O número de rótulos neste caso é um valor subjetivo.

$$\hat{L} = \arg \min_{1 \leq k \leq L_{max}} -\log f(y|\hat{\theta}_k) + a(N)m(k) + b(k, N), \quad (4.1)$$

onde k é o candidato à ordem do modelo, $\hat{\theta}_k$ é a estimativa dos parâmetros por máxima-verossimilhança, L_{max} é a ordem máxima do modelo, N é o número de dados (neste caso, o número de pixels), $m(k)$ é o número de parâmetros do modelo e $a(N)$ e $b(k, N)$ são funções propriamente selecionadas. Neste trabalho, utilizou-se a função log-verossimilhança da equação (2.23), calculada após a última iteração do algoritmo EM/MPM. Assim, a função $\log f(y|\theta_k)$ da equação (4.1) é considerada como $\log f(\mathbf{y}|\mathbf{x}, \hat{\theta}_k)$.

Para a realização dos testes de validação com esses critérios, foi utilizada a imagem sintética apresentada na figura 4.1 (a), que possui sabidamente cinco diferentes classes. Esta imagem foi obtida modificando regiões de uma textura em brilho, contraste e com inserção de ruídos. Outra imagem para teste utilizada foi a da figura 4.1 (b), na qual é mostrado um rosto de um macaco. Nesta não é possível definir ao certo o número de regiões diferentes, já que este valor em uma imagem real depende do ponto de vista do observador. As funções log-verossimilhança resultante da variação de L destas imagens podem ser vistas nas figuras 4.2 (a) e 4.3 (a), respectivamente.

AIC - Critério de Akaike

Akaike [Aka74] foi pioneiro na utilização do princípio ML no problema da seleção da ordem de modelos, aplicando seu estudo em modelos de séries temporais. Reconhecendo que a aplicação direta da função log-verossimilhança leva sempre à escolha da ordem mais alta, foi adicionado um termo de penalidade dependente da estrutura. Assim, no modelo de segmentação EM/MPM, o critério de Akaike é

$$\hat{L} = \arg \min_{1 \leq k \leq L_{max}} \text{AIC}(k),$$

onde

$$\text{AIC}(k) = -\log f(\mathbf{y}|\mathbf{x}, \hat{\boldsymbol{\theta}}_k) + K', \quad (4.2)$$

com K' definido como o número de parâmetros livres (neste caso, o número de parâmetros a estimar). No modelo EM/MPM, tem-se $K' = 2k$, já que existem uma média e uma variância para cada rótulo. Apesar de ter sido muito criticado [Ris78, Kas80], este critério foi utilizado com relativo sucesso em diversos trabalhos [Zha88, She90]. Nas figuras 4.2 (b) e 4.3 (b) apresenta-se o comportamento deste critério na segmentação EM/MPM das figuras 4.1 (a) e (b), respectivamente. Nota-se que a penalidade utilizada é muito pequena em relação à log-verossimilhança, tornando-a ineficaz. Isto é percebido observando que os gráficos das figuras 4.2 (a) e (b) são praticamente simétricos, assim como os das figuras 4.3 (a) e (b). Os valores estimados de L , nesses casos, tendem a ser L_{max} .

MDL - Comprimento de Descrição Mínimo

Desenvolvido por Rissanen [Ris78] e, paralelamente, por Schwarz [Sch78], inspirados no trabalho de Akaike, este critério foi criado procurando anular a natureza tendenciosa a valores altos de L do critério AIC. Ele se baseia na hipótese de que as características do modelo podem ser estudadas por meio do comprimento de dados que o descreve. Desta maneira, sua formulação pode ser descrita como

$$\hat{L} = \arg \min_{1 \leq k \leq L_{max}} \text{MDL}(k),$$

onde

$$\text{MDL}(k) = -\log f(\mathbf{y}|\mathbf{x}, \hat{\boldsymbol{\theta}}_k) + \frac{1}{2}K' \log N, \quad (4.3)$$

com K' definido como no critério AIC. Assim, deve ser escolhida a ordem do modelo que minimiza o comprimento total da descrição compreendida pelos dados e pelos parâmetros. Como visto nas figuras 4.2 (c) e 4.3 (c), esta penalidade também não tem influência prática na curva, que ainda é dominada pela log-verossimilhança.

MGZ - Critério de Merhav-Gutman-Ziv

Este critério é construído utilizando-se o MDL e procura encontrar o ponto em que este começa a decrescer menos. A subestimação e a sobrestimação de L são balanceadas por meio de um parâmetro λ . Desta forma, o critério pode ser descrito como

$$\hat{L} = \arg \min_{1 \leq k \leq L_{max}} [\text{MGZ} \leq \lambda],$$

onde

$$\text{MGZ}(k) = \frac{1}{N}[\text{MDL}(k) - \text{MDL}(L_{max})]. \quad (4.4)$$

A figura 4.2 (d) mostra a curva do critério MGZ correspondente às segmentações da figura 4.1 (a). Utilizando o valor $\lambda = 0.05$ escolhido empiricamente, é possível encontrar o número $\hat{L} = 5$ correto. Entretanto, esse valor é muito dependente da segmentação. Este mesmo λ escolhido na curva MGZ das segmentações da figura 4.1 (b), vista na figura 4.3 (d), resulta na escolha de $\hat{L} = 11$.

WDC - Critério de Won e Derin

Este critério, que tem como objetivo tornar a penalidade compatível com a magnitude da função log-verossimilhança, é dado como

$$\hat{L} = \arg \min_{1 \leq k \leq L_{max}} \text{WDC}(k),$$

onde

$$\text{WDC}(k) = -\log f(\mathbf{y}|\mathbf{x}, \hat{\boldsymbol{\theta}}_k) + N^c K', \quad (4.5)$$

com c obtido experimentalmente. O número correto de rótulos $\hat{L} = 5$ da imagem da figura 4.1 (a) foi corretamente estimado utilizando $c = 0.6$, como visto na figura 4.2 (e). Apesar do bom resultado obtido, a performance do WDC é muito sensível à escolha desta constante, que pode ser diferente para cada imagem. Utilizando este mesmo valor, foi obtida a estimação $\hat{L} = 7$ para a imagem da figura 4.1 (b), como visto na figura 4.3 (e).

Critério do Ajuste da Log-Verossimilhança

Observando que a função log-verossimilhança possui freqüentemente um comportamento exponencial ascendente em relação ao valor de L (semelhante à voltagem em um capacitor sendo carregado), Langan *et al.* [LMZ98] desenvolveram o seguinte critério:

$$\hat{L} = \arg \min_{1 \leq k \leq L_{max}} |0.1 - \exp(-\alpha k)|, \quad (4.6)$$

onde γ e α são escolhidos de forma a minimizar

$$f(\gamma, \alpha) = \log f(\mathbf{y}|\mathbf{x}, \hat{\boldsymbol{\theta}}_k) - \gamma[1 - \exp(-\alpha k)] \quad (4.7)$$

utilizando a técnica dos mínimos quadrados aplicada a funções não-lineares. O ponto principal deste critério é que o modelo exponencial suaviza a função log-verossimilhança, que tende a formar ruídos dadas as condições iniciais, as aproximações e os erros da modelagem da segmentação.

A minimização da equação (4.7) busca encontrar os parâmetros γ e α que ajustam uma curva semelhante ao comportamento desta função. Para encontrar o “joelho” da curva, que representa a estimativa da ordem do modelo, calcula-se o ponto onde esta alcança 90% de ascensão como na equação (4.6), aproximando k pelo valor inteiro mais próximo. Nas figuras 4.2 (f) e (g) são apresentados os resultados obtidos no teste realizado para a imagem da figura 4.1 (a), apresentando o ajuste com a log-verossimilhança normalizada e a curva para estimação de L , respectivamente. No caso do teste da imagem da figura 4.1 (b), os resultados podem ser vistos nas figuras 4.3 (f) e (g).

De fato, a utilização deste critério elimina muitos dos problemas ocorridos nos critérios anteriormente apresentados. Primeiramente, a magnitude da função log-verossimilhança se torna previsível, já que esta foi normalizada. O ajuste de uma função contínua semelhante torna sua análise mais fácil. Entretanto, o critério ainda parte do princípio que a log-verossimilhança tem sempre o mesmo comportamento, o que não foi observado em todos os experimentos.

Figura 4.2: (a) Função log-verossimilhança e a aplicação, à imagem da figura 4.1 (a), dos critérios (b) AIC, (c) MDL, (d) MGZ (com $\lambda = 0.05$) e (e) WDC (com $c = 0.6$). (f) Ajuste da log-verossimilhança e (g) aplicação do critério de ajuste. Em todos os gráficos, o eixo horizontal representa os diferentes valores de L .

Figura 4.3: (a) Função log-verossimilhança e a aplicação, à imagem da figura 4.1 (b), dos critérios (b) AIC, (c) MDL, (d) MGZ (com $\lambda = 0.05$) e (e) WDC (com $c = 0.6$). (f) Ajuste da log-verossimilhança e (g) aplicação do critério de ajuste. Em todos os gráficos, o eixo horizontal representa os diferentes valores de L .

4.2 Distância entre Distribuições

Neste trabalho foi desenvolvido um estudo sobre a possibilidade da análise das distribuições dos rótulos para auxiliar no processo de validação de classes da segmentação EM/MPM. Essas distribuições, como apresentado na seção 2.1.2, são definidas como uma função gaussiana. Esta definição sugere que o histograma dos pixels com mesmo rótulo deve ter o formato de uma distribuição normal, o que de fato é observado na grande maioria dos casos. Esta abordagem será chamada de **distância entre distribuições**.

Para encontrar critérios com esta abordagem, observa-se a imagem sintética apresentada na figura 4.1 (a) que possui sabidamente cinco diferentes classes. Nas figuras 4.4 a 4.10 podem ser observadas, para cada segmentação realizada pela implementação do algoritmo EM/MPM descrito na seção 3.1, com número de rótulos entre 2 e 8, as distribuições estimadas pelo algoritmo EM juntamente com o histograma das L diferentes classes classificadas. Detalhes sobre os parâmetros iniciais destes resultados serão discutidos no capítulo 5. A observação dos gráficos leva a duas conclusões no sentido de encontrar o número real de classes da imagem:

1. Havendo alguma distribuição teórica muito diferente do histograma observado do rótulo, é provável que a imagem esteja subsegmentada. Isto é percebido ao comparar-se os gráficos das figuras 4.4 a 4.6 com os das figuras 4.7 a 4.10. Apenas nesses primeiros é possível observar que ao menos um dos histogramas de algum rótulo possui mais que um pico, diferenciando-se muito da distribuição normal correspondente.
2. O aparecimento de distribuições gaussianas muito semelhantes está relacionado com a sobresegmentação de uma determinada região da imagem. Nestes casos, é provável que o número de classes correto seja menor do que o escolhido, já que a região pode ser representada por apenas um rótulo. Isto pode ser observado nos gráficos das figuras 4.8 a 4.10, onde existem duas ou mais distribuições extremamente próximas umas das outras, ou seja, com valores de média e variância semelhantes.

Para medir a diferença entre o histograma observado e a distribuição teórica, uma técnica interessante é a **estatística de Kolmogorov-Smirnov**. Já para medir a diferença entre duas distribuições um método possível é o cálculo da **distância de Bhattacharyya**. Essas duas medidas, assim como suas aplicações aos resultados mostrados nas figuras 4.4 a 4.10, serão apresentadas nas seções 4.2.1 e 4.2.2, respectivamente. Nestas seções, também serão mostrados os resultados obtidos com a imagem do rosto de macaco da figura 4.1 (b), cujas segmentações podem ser vistas na figura 4.11.

4.2.1 Estatística de Kolmogorov-Smirnov

A estatística de Kolmogorov-Smirnov (KS) tem como objetivo analisar se dados observados são compatíveis com uma determinada distribuição teórica. Isso é feito calculando-se a maior diferença entre as funções de distribuição empírica e teórica. Desta forma, a estatística KS é dada por

$$KS = \sup_x |F(x) - S(x)| \quad \forall x, \quad (4.8)$$

onde $S(x)$ e $F(x)$ são as as funções de distribuição empírica e teórica acumuladas, respectivamente.

Figura 4.4: (a) Resultado da segmentação da imagem da figura 4.1 (a) com $L = 2$ e (b) gráfico com os histogramas de cada classe e as respectivas distribuições gaussianas ajustadas.

Figura 4.5: (a) Resultado da segmentação da imagem da figura 4.1 (a) com $L = 3$ e (b) gráfico com os histogramas de cada classe e as respectivas distribuições gaussianas ajustadas.

Figura 4.6: (a) Resultado da segmentação da imagem da figura 4.1 (a) com $L = 4$ e (b) gráfico com os histogramas de cada classe e as respectivas distribuições gaussianas ajustadas.

Figura 4.7: (a) Resultado da segmentação da imagem da figura 4.1 (a) com $L = 5$ e (b) gráfico com os histogramas de cada classe e as respectivas distribuições gaussianas ajustadas.

Figura 4.8: (a) Resultado da segmentação da imagem da figura 4.1 (a) com $L = 6$ e (b) gráfico com os histogramas de cada classe e as respectivas distribuições gaussianas ajustadas.

Figura 4.9: (a) Resultado da segmentação da imagem da figura 4.1 (a) com $L = 7$ e (b) gráfico com os histogramas de cada classe e as respectivas distribuições gaussianas ajustadas.

Figura 4.10: (a) Resultado da segmentação da imagem da figura 4.1 (a) com $L = 8$ e (b) gráfico com os histogramas de cada classe e as respectivas distribuições gaussianas ajustadas.

Figura 4.11: Segmentações da imagem da figura 4.1 (b) com $L =$ (a) 2, (b) 3, (c) 4, (d) 5, (e) 6, (f) 7, (g) 8, (h) 9, (i) 10, (j) 11, (l) 12, (m) 13, (n) 14, (o) 15 e (p) 16.

A utilização desta estatística na validação é exemplificada na figura 4.12. As curvas representam as distribuições acumuladas empíricas e as distribuições teóricas dos rótulos da segmentação com $L = 3$ apresentada na figura 4.5. Nota-se que as classes mais escura e mais clara (com médias próximas de 0 e de 255, respectivamente) exibem diferenças mais perceptíveis entre as curvas teórica e observada, resultando em valores mais altos de KS. Entretanto, a classe intermediária está corretamente segmentada, já que as curvas teórica e observada são bem próximas, resultando em um pequeno valor de KS, normalmente menor que 0.04.

Figura 4.12: Estatística de Kolmogorov-Smirnov para a segmentação da imagem da figura 4.1 (a) com $L = 3$. Supõe-se que dois dos rótulos podem ser subdivididos em mais classes, já que suas distribuições são diferentes dos histogramas observados.

Foi desenvolvido neste trabalho um critério simples para justificar a possibilidade do uso desta abordagem na validação de classes para o método EM/MPM. Primeiramente, para cada segmentação com diferentes valores de L , calcula-se o valor de KS de todos os rótulos para, em seguida, escolher o maior entre eles. Deve ser escolhido o menor valor de L para o qual o maior valor KS é menor do que um dado λ . Formalizando,

$$\hat{L} = \arg \min_{1 \leq L \leq L_{max}} [KS^*(L) \leq \lambda], \quad (4.9)$$

onde

$$KS^*(L) = \max_{1 \leq k \leq L} KS_k.$$

Na figura 4.13 é possível observar o comportamento de $KS^*(L)$ nas segmentações das figuras 4.4 a 4.10. Com $L=2, 3$ e 4 percebe-se que existe algum rótulo com valor de KS alto, ou seja, com a distribuição observada mal representada pela distribuição teórica. Intuitivamente, esse fato leva a pensar que há necessidade de considerar mais classes na imagem. Com $L=5, 6, 7$ e 8 todas as curvas estão bem representadas pelas distribuições, já que o número ótimo de rótulos foi atingido. Para as segmentações da figura 4.11, a aplicação deste critério pode ser vista na figura 4.14. Observa-se que o comportamento da função não é tão bem comportado quanto no caso da imagem sintética. Há dois mínimos

Figura 4.13: Comportamento do critério KS^* na imagem da figura 4.1 (a). A partir de $L = 5$ observa-se que não existem mais rótulos a serem subdivididos. Escolhendo um λ pertinente, o critério retorna o número de classes correto.

locais para $L=8$ e 12 , comportamento que reflete o caráter subjetivo na definição de L .

4.2.2 Distância de Bhattacharyya

A distância de Bhattacharyya [Bha43] é uma importante medida de separabilidade entre duas distribuições. Como descrito por Fukunaga [Fuk90], para distribuições normais,

$$B = \frac{1}{4} \left\{ \frac{(\mu_2 - \mu_1)^2}{\sigma_1^2 + \sigma_2^2} \right\} + \frac{1}{4} \log \left\{ \frac{1}{4} \left(\frac{\sigma_1^2}{\sigma_2^2} + \frac{\sigma_2^2}{\sigma_1^2} + 2 \right) \right\}, \quad (4.10)$$

onde μ_1, σ_1^2 e μ_2, σ_2^2 são os parâmetros de média e variância das distribuições 1 e 2, respectivamente. O primeiro e o segundo termo desta equação desaparecem quando $\mu_1 = \mu_2$ e $\sigma_1^2 = \sigma_2^2$, respectivamente. Desta forma, o primeiro termo representa a diferença entre as médias e, o segundo, a diferença entre as variâncias.

Na validação de classes do método EM/MPM, essa medida pode auxiliar na busca de classes sobresegmentadas, ou seja, que podem ser representadas por um só rótulo. Na figura 4.8 (a) por exemplo, é possível perceber que a região de fundo está segmentada em duas classes, sendo que existe apenas uma textura. Isso é visto também em (b), onde existem duas distribuições muito próximas. O valor da distância de Bhattacharyya entre elas, portanto, deve ser baixo.

Um possível critério utilizando este conceito foi estudado e desenvolvido neste trabalho. Primeiramente calcula-se, para cada segmentação com um valor de L diferente, o valor B entre todos os possíveis pares de distribuições, escolhendo o mínimo entre eles. O valor L estimado deve ser o maior entre todos onde o máximo B seja maior que um dado λ . Desta forma,

$$\hat{L} = \arg \max_{1 \leq L \leq L_{max}} [B^*(L) \geq \lambda], \quad (4.11)$$

onde

Figura 4.14: Comportamento do critério KS^* na imagem da figura 4.1 (b).

$$B^*(L) = \min_{i,j \in \{1, \dots, L\} | i \neq j} B_{i,j}.$$

Na figura 4.15 é apresentado o resultado da aplicação deste critério aos testes com a imagem da figura 4.1 (a). Com $L=6, 7$ e 8 percebe-se que existem pelo menos dois rótulos próximos e, por isso, com valor B pequeno. Essas estimativas devem ser descartadas, já que existem mais rótulos do que o necessário. Entretanto, com $L=2, 3, 4$ e 5 não existe nenhum par de distribuições que se confundem. Escolhe-se o maior valor entre essas estimativas, ou seja, $\hat{L} = 5$. Para a imagem da figura 4.1 (b), o comportamento da função $B^*(L)$ pode ser visto na figura 4.16. Para o valor de λ escolhido, o número estimado de classes foi $\hat{L} = 8$.

4.3 Considerações

Após analisar os critérios de validação de classes apresentados na seção 4.1, alguns aspectos se mostraram relevantes. A função log-verossimilhança, por exemplo, apresentou um comportamento que de fato sugere seu uso para a busca do número ideal de rótulos. Entretanto, sua magnitude é muito dependente da imagem e difícil de ser prevista. Um outro ponto importante é que, nestes critérios, assume-se que é possível calcular o valor da função log-verossimilhança para $L = 1$, ou seja, para apenas um rótulo. O comportamento acumulativo da função log-verossimilhança sugere que a estimativa correta para o número de rótulos será correto para valores de L igual ou superior a 2. Isto significa que o gráfico resultante da função log-verossimilhança para $L \geq 1$ sempre exibirá um salto de $L = 1$ para $L = 2$, podendo se estabilizar a partir de então, excluindo a validação de classes para $L = 1$. No entanto, não é possível estimar a log-verossimilhança para $L = 1$ no método EM/MPM implementado, sendo que o processo de segmentação assume que o número mínimo de rótulos é igual a 2. Desta forma, exclui-se a possibilidade de se obter uma validação correta, quando esta for $L = 2$.

Outro problema encontrado foi a aplicação dos critérios apresentados na seção 4.1 para um número muito grande de rótulos, ou seja, com L_{max} muito alto. Nessas situações, os gráficos tendem a apresentar mais de um mínimo local à medida que o número de rótulos

Figura 4.15: Comportamento do critério B^* para a imagem da figura 4.1 (a). A partir de $L = 6$ observa-se que existem rótulos com distribuições muito parecidas, sendo que a imagem fica sobresegmentada.

Figura 4.16: Comportamento do critério B^* para a imagem da figura 4.1 (b).

aumenta, sugerindo assim a existência de múltiplos valores de validação corretos. Isto é percebido, por exemplo, observando a função de log-verossimilhança da figura 4.3 (a). Quando $L = 2, \dots, 9$, o gráfico tem um comportamento de forma a se estabilizar, ou seja, a cada aumento de L , a função cresce mais lentamente. Entretanto há um aumento mais significativo deste valor entre $L = 9$ e 10 , estabilizando novamente quando $L = 16$.

Por fim, a análise das distribuições se mostrou uma abordagem mais intuitiva para a validação de classes para o método EM/MPM. Apesar da necessidade da escolha do parâmetro de limiar λ , o comportamento das curvas dos critérios de validação pela estatística de Kolmogorov-Smirnov e distância de Bhattacharyya foi de fato condizente com o esperado na imagem sintética. Na imagem do babuíno, as irregularidades nas curvas ocorrem devido à subjetividade da existência de um valor real para o número de classes. Desta forma, acredita-se que esta é uma abordagem realmente interessante que deve ser mais profundamente estudada.

Resultados

Os experimentos realizados com as técnicas apresentadas neste trabalho demonstraram que estas são eficazes para diferentes tipos de imagens. Desta forma, podem ser aplicadas a uma extensa gama de problemas, como por exemplo na segmentação de imagens médicas e de satélite. Além disso, foi desenvolvida uma extensão do método EM/MPM para segmentação de malhas. Os resultados obtidos com esta nova implementação se mostraram bastante promissores.

Em todos os testes, as variâncias iniciais utilizadas foram definidas como $\sigma_k^2 = 700$, para $k = 1, \dots, L$. A escolha deste valor pouco influi na segmentação final ou no tempo de convergência, não sendo necessário, portanto, uma discussão mais detalhada a seu respeito. O número de visitas por pixel foi mantida como $\mathcal{T} = 3$ em todos os testes, dado que a utilização de valores maiores implica em um grande aumento do tempo de processamento por iteração sem muito ganho de qualidade na segmentação final.

Os testes foram realizados com a utilização da estimação das médias iniciais μ_k para $k = 1, \dots, L$ dada pela equação (2.31) e por meio do algoritmo proposto na seção 3.2. Para facilitar a leitura, estes dois métodos serão chamados de **EMIN** (Estimação de Médias Iniciais Normal) e **EMIL** (Estimação de Médias Iniciais por Limiarização), respectivamente.

Neste capítulo serão apresentados alguns dos resultados obtidos com os métodos descritos neste trabalho. Nas seções 5.1, 5.2 e 5.3 serão mostrados os testes realizados com imagens sintéticas, médicas e de cenas naturais, respectivamente. Por fim, na seção 5.4, serão expostos a modelagem e os resultados obtidos na extensão do método EM/MPM para segmentação de malhas. Para efeito de comparação, alguns resultados da segmentação com *Clustering K-Means* [WS90], que utiliza texturas extraídas de matrizes de co-ocorrência de Haralick [Har79] e *Run Length* [Gal75], serão também fornecidos.

5.1 Imagens Sintéticas

Para iniciar a discussão sobre os resultados obtidos, foram realizados alguns testes com imagens sintéticas. Primeiramente, foi utilizada a imagem da figura 4.1 (a), apresentada no capítulo 4. Como dito anteriormente, ela contém cinco regiões com diferentes micro-texturas, criadas a partir da modificação do brilho, do contraste e da inserção de ruídos.

Na figura 5.1 é mostrada a segmentação com o algoritmo *K-Means* para $L = 5$. Foram empregadas cinco características de textura, extraídas das matrizes de co-ocorrência de Haralick. Embora quatro regiões majoritárias tenham sido identificadas, o algoritmo não foi capaz de prover uma segmentação uniforme. A segmentação EM/MPM foi realizada em quatro diferentes contextos:

1. Utilizando EMIN, com o parâmetro $\beta = 1.2$. As figuras 5.2 (a), (b) e (c) representam os resultados para 1, 100 e 170 iterações, respectivamente.
2. Utilizando EMIL, com o parâmetro $\beta = 1.2$. As figuras 5.2 (d), (e) e (f) representam os resultados para 1, 20 e 45 iterações, respectivamente.
3. Utilizando EMIN e a técnica de *annealing*, com parâmetros $\beta_0 = 0$ e $I = 0.045$. As figuras 5.2 (g), (h) e (i) representam os resultados para 1, 15 e 35 iterações, respectivamente.
4. Utilizando EMIL e a técnica de *annealing*, com parâmetros $\beta_0 = 0.5$ e $I = 0.045$. As figuras 5.2 (j), (l) e (m) representam os resultados para 1, 10 e 25 iterações, respectivamente.

Apesar da semelhança no resultado final dos diferentes contextos, nota-se que o número de iterações necessárias para uma boa segmentação variou. Tanto o método de estimação inicial das médias por limiarização quanto a técnica de *annealing* atuaram de maneira a aumentar consideravelmente a velocidade de convergência, pois demandaram um número menor de iterações para obter um resultado satisfatório. Além disto, o uso conjunto destas duas abordagens se mostrou ainda mais vantajoso.

Para analisar o efeito da estimação inicial das médias no algoritmo EM/MPM, deve-se comparar a segmentação da primeira iteração sem e com o uso do método da limiarização, dadas nas figuras 5.2 (a) e 5.2 (d), respectivamente. Nota-se que o uso do método da limiarização resulta em uma melhor segmentação inicial, permitindo uma convergência mais rápida. Neste caso, o algoritmo gasta menos tempo procurando as distribuições corretas dos rótulos, já que elas foram previamente melhor estimadas. Nos gráficos das figuras 5.3 (a) e (b) é mostrado o histograma final dos pixels de cada rótulo juntamente com as médias estimadas sem e com limiarização, respectivamente. Nota-se que no segundo caso as médias estão de fato mais próximas aos seus valores reais.

O efeito do *annealing* pode ser percebido comparando-se os resultados das figuras 5.2 (a) a (c) e 5.2 (g) a (i). Nestes primeiros, observa-se que a falha da primeira segmentação se mantém por bastante tempo. Este problema, porém, não ocorre com o uso do *annealing*. Nesta abordagem, os pixels não se mantêm fixos nas primeiras iterações, pois o valor β de interação espacial é baixo. Este fato permite ao método EM encontrar, já no início do processo, estimações mais precisas das distribuições para, só então, com o aumento de β , unir pixels vizinhos com mesmo rótulo.

Outros testes do algoritmo EM/MPM em imagens sintéticas foram realizados. As imagens das figuras 5.4 (a) e 5.5 (a) foram feitas com a combinação de texturas do álbum de Brodatz [Bro66]. As segmentações finais com dois rótulos podem ser vistas nas figuras 5.4 (b) e 5.5 (b), respectivamente. No trabalho de Comer [Com95], resultado semelhante à figura 5.4 (a) foi conseguido somente com o uso de uma abordagem multiresolução. Neste trabalho o uso do *annealing* permitiu utilizar valores altos de β nas últimas iterações, deixando o resultado mais robusto e dispensando o processo multiresolução.

Por fim, um último experimento foi realizado com a imagem sintética da figura 5.6 (a), composta por três texturas do álbum de Brodatz [Bro66]. Nota-se que essas são macrotexturas, ou seja, texturas com bastante granulosidade que só são reconhecidas por uma visão menos detalhada. Isso é comprovado observando o resultado da segmentação EM/MPM desta imagem, mostrado na figura 5.6 (b), onde as regiões não se encontram corretamente definidas. Em (c) é ilustrada a segmentação de *K-Means* multiresolução com características de *Run Length*. A abordagem multiresolução se mostra mais eficiente neste caso.

Nas imagens das figuras 5.7 (a) a (d) são mostrados quatro níveis de resolução da imagem da figura 5.6 (a). Nota-se que, nos maiores níveis (resoluções baixas), as texturas apresentam menos detalhes. Utilizando a segmentação EM/MPM multiresolução proposta na seção 2.2 obtém-se os resultados das figuras 5.6 (e) a (h). O resultado do nível mais baixo, que possui a mesma resolução da imagem original, se mostra bem próximo do esperado, comprovando a eficácia desta abordagem.

Figura 5.1: Segmentação por *K-Means* da imagem da figura 4.1 (a) com características de Haralick, com $L = 5$.

Figura 5.2: Segmentação EM/MPM da imagem da figura 4.1 (a) com $L = 5$ nos seguintes contextos: (a) EMIN e $\beta = 1.2$, para $\mathcal{P} = 1$, (b) 100 e (c) 170. (d) EMIL e $\beta = 1.2$, para $\mathcal{P} = 1$, (e) 20 e (f) 45. (g) EMIN, *annealing* com $\beta_0 = 0$ e $I = 0.045$, para $\mathcal{P} = 1$, (h) 15 e (i) 35. (j) EMIL, *annealing* com $\beta_0 = 0.5$ e $I = 0.045$, para $\mathcal{P} = 1$, (l) 10 e (m) 25.

Figura 5.3: (a) Médias estimadas como na equação (2.31) (EMIN) e (b) por meio da limiarização proposta na seção 3.2 (EMIL).

Figura 5.4: (a) Imagem sintética composta por texturas do álbum de Brodatz [Bro66] e sua (b) segmentação EM/MPM com EMIL, $L = 2$, $\beta_0 = 0$, $I = 0.02$, $\beta_{max} = 1.75$ e $\mathcal{P} = 230$.

Figura 5.5: (a) Imagem sintética composta por texturas do álbum de Brodatz [Bro66] e sua (b) segmentação EM/MPM com EMIL, $L = 2$, $\beta_0 = 1$, $I = 0.25$, $\beta_{max} = 20$ e $\mathcal{P} = 200$.

Figura 5.6: (a) Imagem sintética composta por três texturas do álbum de Brodatz [Bro66], (b) segmentação EM/MPM com EMIL, $L = 3$, $\beta_0 = 0$, $I = 0.025$ e $\mathcal{P} = 250$ e (c) segmentação com multiresolução e correção de bordas.

Figura 5.7: Imagens nos níveis $n = (a)$ 1, (b) 2, (c) 3 e (d) 4 de resolução. Segmentação EM/MPM multiresolução com $M = 4$, $L = 3$, $\beta_0^{(n)} = 1.2$, $I^{(n)} = 0.025$ e $\alpha^{(n)} = 2$ para todo n nos níveis (e) 1, (f) 2, (g) 3 e (h) 4.

5.2 Imagens Médicas

A imagem da figura 5.8 (a) é uma imagem de mamografia com as características típicas de um fibroadenoma, tumor benigno comum em jovens com idade fértil. A segmentação em (b) foi conseguida utilizando a abordagem multiresolução. Nota-se que a região do nódulo foi corretamente separada do plano de fundo e do tecido saudável.

Na figura 5.9 pode ser visualizada a imagem da tomografia axial de um cérebro e suas segmentações EM/MPM em diferentes contextos. Em (b) foi feita a segmentação com $L = 4$, EMIN e β constante. Nota-se que, neste caso, as regiões não foram bem definidas. Entretanto, com a introdução da EMIL em (c), do *annealing* em (d) e destes dois métodos simultaneamente em (e), foram obtidas segmentações mais satisfatórias.

Como percebido nessas figuras, o algoritmo de segmentação não foi eficaz na separação da região do plano de fundo, que possui muitos pixels pretos (com valor $y_s = 0$). Nos casos onde a textura a ser segmentada tem em sua grande maioria pixels com mesma cor, o método EM tende a produzir uma variância com valor muito próxima a zero para o rótulo correspondente. Desta forma, mesmo com uma pequena mudança no nível de cinza de um pixel, este é classificado com um rótulo diferente.

Para corrigir este problema, realizou-se uma nova segmentação da imagem levando-se em consideração apenas os pixels com níveis de cinza maiores do que 6, ou seja, para todo $s|y_s \geq 7$, sendo que aos pixels restantes foi atribuído o rótulo 0 (nulo). Os resultados com $L = 3, 4$ e 5 podem ser vistos nas figuras 5.9 (f), (g) e (h), respectivamente. Esta abordagem da escolha de uma faixa de nível de cinza a ser segmentada pode ser vista como uma pré-segmentação por limiarização. Além de poder melhorar os resultados, o desempenho do algoritmo aumenta consideravelmente, já que muitos pixels são ignorados.

Figura 5.8: (a) Imagem de mama com um fibroadenoma retirado do *site* Surgical-Tutor [ST06] e sua (b) segmentação EM/MPM multiresolução com $M = 4$, $\alpha = 3.5$, EMIN, $L = 3$ e $\beta = 1.2$.

Figura 5.9: (a) Imagem digital de tomografia axial de um cérebro. Para todos os testes, foram realizadas $\mathcal{P} = 100$ iterações do algoritmo EM/MPM. (b) Resultados com $L = 4$, $\beta = 1.2$, EMIN e (c) EMIL. (d) Resultados com $L = 4$, *annealing* com $\beta_0 = 0$, $I = 0.025$, $\beta_{max} = 1.2$, EMIN e (e) EMIL. (f) Resultados da segmentação restrita aos pixels $s|y_s \geq 7$, com EMIL, *annealing* com $\beta_0 = 0$, $I = 0.025$, $\beta_{max} = 1.2$, $L = 3$ (g) $L = 4$ e (h) $L = 5$.

5.3 Cenas Naturais

A figura 5.10 é uma foto de satélite de um trecho do rio Amazonas. O resultado da figura 5.11 (a) para a segmentação *K-Means* com duas características de co-ocorrência de Haralick e $L = 5$ mostra que, com exceção da região ocupada pelo rio, o algoritmo não é capaz de retornar áreas homogêneas. A segmentação EM/MPM foi realizada em diferentes contextos, com $L = 5$: uso de EMIN, EMIL e EMIL com *annealing*, sendo os resultados mostrados nas figuras 5.11 (b), (c) e (d), respectivamente. Mais uma vez pode-se observar que as estimativas iniciais das médias em muito influenciam o resultado final da segmentação. Além disso, a utilização do *annealing* neste caso foi importante para gerar um resultado mais estável, pois o valor de β nas últimas iterações é alto o suficiente para diminuir a ocorrência de mínimos locais.

Uma aplicação do método EM/MPM proposta por Balan [Bal03], foi na segmentação de imagens de ninhais de aves para uma posterior contagem. O censo de pássaros em uma região é utilizado, por exemplo, para detectar eventuais distúrbios que ocorrem no meio ambiente, podendo por isso ser utilizado como monitoramento biológico [LLL00]. Na imagem da figura 5.12 (a) é apresentada a foto de um bando de aves. Observa-se que a maioria dos pixels pertencem a regiões da vegetação e da água, sendo normalmente mais escura. Por isso, foi aplicado o algoritmo de segmentação EM/MPM restrito apenas aos pixels $s|s \geq 80$. O resultado em (b) mostra que as aves foram corretamente separadas dos outros elementos da imagem, tais como seus reflexos, os galhos e outros. Apesar da ocorrência de alguns ruídos, como visto na figura com o rótulo separado em (c), todas as aves foram identificadas.

Para observar o efeito da multiresolução, foi realizado um experimento utilizando a imagem da figura 5.13 (a). Na segmentação EM/MPM dessa imagem vista em (b), percebe-se que os tijolos são separados uns dos outros. Já com o resultado do uso da abordagem multiresolução em (c), toda a parede é vista como uma só região. Isso ocorre pois, com a diminuição da resolução, a textura formada pela parede de tijolos é vista como uma só.

Figura 5.10: Imagem de satélite de um trecho do rio Amazonas.

(a)

(b)

(c)

(d)

Figura 5.11: Segmentações da imagem da figura 5.10. *a)* Segmentação por *K-Means*. *(b)* Segmentação EM/MPM com $\mathcal{P} = 250$ e $L = 5$ e *(b)* resultados com $\beta = 1.2$, EMIN e *(c)* EMIL e *(d)* e com EMIL, $\beta_0 = 0$ e $I = 0.05$.

(a)

(b)

(c)

Figura 5.12: (a) Fotografia digital de um bando de aves e sua (b) segmentação EM/MPM dos pixels $s|y_s \geq 80$, com EMIL, $L = 3$, $\beta_0 = 0$, $I = 0.05$ e $\mathcal{P} = 200$. (c) Pixels com a classe das aves.

(a)

(b)

(c)

Figura 5.13: (a) Fotografia digital de uma parede. Segmentações EM/MPM com EMIL, $L = 6$, $\beta_0 = 0$, $I = 0.04$ e $\mathcal{P} = 120$, (b) sem e (c) com abordagem multiresolução para $M = 2$.

5.4 Malhas

Uma outra contribuição deste trabalho foi a extensão do método EM/MPM de segmentação de imagens para a segmentação de malhas de duas e três dimensões. A grande diferença entre a estrutura de malhas e de imagens consiste na geometria dos elementos. No caso das imagens, todos os pixels são iguais em forma, simetria e vizinhança. Já as malhas possuem elementos de diferentes tamanhos, formatos e vizinhanças, sendo que cada vizinho de um elemento pode possuir uma maior ou menor adjacência com este. Nas figuras 5.14 (a) e (b) é apresentado o esquema geral de vizinhança em uma malha 2D (conjunto de triângulos) e 3D (conjunto de tetraedros), respectivamente, formando as cliques mostradas em (c).

Figura 5.14: Vizinhanças para malhas no caso (a) 2D e (b) 3D, com suas (c) cliques correspondentes.

Na extensão do método EM/MPM para segmentação de malhas, cada vizinho de um elemento pode ter um peso diferente dependendo da geometria do espaço que compartilham. Com isso, a equação (2.5) pode ser reformulada para a seguinte forma:

$$t(x_r, x_s) = \begin{cases} 0, & \text{se } x_r = x_s, \\ \frac{a_{s,r}}{p_s}, & \text{se } x_r \neq x_s, \end{cases} \quad (5.1)$$

onde, em 2D, $a_{s,r}$ é o comprimento da aresta que divide os triângulos s e r e p_s é o perímetro do triângulo s . Já no caso 3D, $a_{s,r}$ pode ser a área que separa os tetraedros s e r , e p_s a área superficial do tetraedro s .

Outro fator a ser alterado do modelo original para segmentação de imagens está no algoritmo EM. No caso de uma imagem, o valor do nível de cinza de cada pixel tem a mesma importância. Entretanto, nas malhas, um triângulo ou tetraedro pode ter maior ou menor peso que outro, dependendo de sua geometria. Para acrescentar esta informação, a estimação de θ , representada pelas equações (2.25) a (2.27), foi reformulada para

$$\mu_k(p) = \frac{1}{N_k(p)} \sum_{s=1}^N y_s P(X_s = k | \mathbf{Y} = \mathbf{y}, \theta(p-1)) \times A_s \quad (5.2)$$

e

$$\sigma_k^2(p) = \frac{1}{N_k(p)} \sum_{s=1}^N (y_s - \mu_k(p))^2 P(X_s = k | \mathbf{Y} = \mathbf{y}, \theta(p-1)) \times A_s, \quad (5.3)$$

onde

$$N_k(p) = \sum_{s=1}^N P(X_s = k | \mathbf{Y} = \mathbf{y}, \boldsymbol{\theta}(p-1)) \times A_s, \quad (5.4)$$

e A_s é a área do triângulo ou volume do tetraedro s . Este valor representa o tamanho do elemento, que dará seu peso no cálculo da média e variância do rótulo k .

Os experimentos com essa nova formulação foram realizados no contexto do trabalho de Alex J. Cuadros-Vargas [CV06], aluno de doutorado do ICMC-USP¹ que desenvolve um sistema para geração de malhas volumétricas a partir de imagens. Para analisar a eficácia do método, os resultados foram comparados com o uso de uma técnica também proposta em seu trabalho, baseada no crescimento de regiões e na vizinhança dos elementos. Para facilitar a leitura, esta técnica será chamada **CRV**.

Para todos os testes, as variâncias dos rótulos foram limitadas ao valor mínimo de 50 durante a segmentação, isto é, $\sigma_k^2 > 50$, $\forall k = 1, \dots, L$. Desta forma, evitou-se o aparecimento de classes com distribuições limitadas a uma faixa muito pequena do histograma. Foram também utilizadas as técnicas de EMIL e de *annealing*, com $L = 3$.

Na figura 5.15 (a) é apresentada uma imagem bidimensional de um pé humano. Nota-se que esta possui texturas e detalhes não muito bem definidos, sendo que a malha gerada a partir desta também possui estas características. O resultado mostrado em (b) foi obtido com a técnica CRV. Comparando esta com figura em (c), nota-se que com o método EM/MPM para segmentação de malhas obteve-se um resultado mais satisfatório, com bordas e regiões mais bem definidas.

Um primeiro exemplo da segmentação de uma malha tridimensional pode ser visto na figura 5.16. Em (a) e (b) são mostradas uma das fatias bidimensionais e a renderização volumétrica, respectivamente, de um grão de pólen. Observa-se uma grande diferença entre a segmentação por CRV e por EM/MPM, apresentadas em (c) e (d), respectivamente. Nota-se claramente que a superfície do resultado da segunda abordagem possui um comportamento muito mais suave e próximo do real.

Por fim, a figura 5.17 mostra um exemplo de imagem tridimensional com estruturas internas: o crânio de um macaco. Em (a) e (b) podem ser vistas uma das fatias bidimensionais e a renderização volumétrica resultante. Em (c) e (d) são mostrados os resultados da segmentação com CRV e com EM/MPM, respectivamente. Mais uma vez o resultado da extensão EM/MPM para malhas se mostrou mais interessante, já que nesta abordagem as estruturas internas foram melhor definidas.

¹Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo

(a)

(b)

(c)

Figura 5.15: (a) Imagem da fatia bidimensional de um pé humano e suas segmentações por (b) CRV e (c) EM/MPM com EMIL, $L = 3$, $\mathcal{P} = 100$, $\beta_0 = 0$, $I = 0.05$ e $\beta_{max} = 5$.

(a)

(b)

(c)

(d)

Figura 5.16: (a) Fatia bidimensional de um grão de pólen, (b) sua renderização volumétrica resultante e as segmentações por (c) CRV e (d) EM/MPM com EMIL, $L = 2$, $\mathcal{P} = 100$, $\beta_0 = 0$, $I = 0.05$ e $\beta_{max} = 5$.

(a)

(b)

(c)

(d)

Figura 5.17: (a) Fatia bidimensional do crânio de um macaco, (b) sua renderização volumétrica resultante e as segmentações por (c) CRV e (d) EM/MPM com EMIL, $L = 2$, $\mathcal{P} = 200$, $\beta_0 = 0$, $I = 0.0333$ e $\beta_{max} = 30$.

Conclusões e Trabalhos Futuros

Neste trabalho foram apresentados o estudo e a implementação do método EM/MPM de segmentação de imagens, bem como sua abordagem multiresolução. Também foram propostas as técnicas de estimação inicial de parâmetros de médias do modelo por limitização e de *annealing*, para diminuir a quantidade de parâmetros a serem previamente definidos e melhorar o resultado da segmentação final. Além disso, foram apresentados alguns métodos para estimar o número ótimo de classes na imagem e uma extensão da modelagem estocástica para a segmentação de malhas de duas e três dimensões.

Assim como nos trabalhos de Comer e Delp [Com95, CD00], a implementação do método EM/MPM se mostrou eficaz para a segmentação de imagens com microtexturas. O campo aleatório de Markov é, de fato, uma abordagem interessante para modelar a dependência espacial dos pixels que formam uma textura. O autor acredita, entretanto, que o algoritmo pode ainda ser modificado de forma a obter melhores resultados. A ordem de visita dos pixels no formato *raster scan* do amostrador de Gibbs, por exemplo, pode ser substituída pelo método de Hilbert-Peano [FPD01], uma técnica recursiva para percurso em uma grade 2D. Outra sugestão do autor é desenvolver a modelagem para um vetor de características de cada pixel ao invés do uso exclusivo do nível de cinza. Este novo modelo poderia ser aplicado, por exemplo, em imagens coloridas.

O uso de modelos não gaussianos (que se mostra insatisfatório em alguns casos) para as distribuições dos rótulos, pode também ser experimentado. O modelo tradicional EM/MPM, proposto por Comer, não é capaz de separar duas classes que possuam a mesma média e variância, mesmo quando os pixels pertencentes a qualquer uma das classes possuam organização espacial distintas. Neste trabalho, porém, este problema foi contornado com a funcionalidade de multiresolução. Diminuindo a resolução, as médias das texturas permanecem inalteradas, mas as variâncias se modificam. Desta forma, as texturas se diferenciam de maneira suficiente para permitir uma segmentação correta.

A utilização de uma pré-segmentação como estimativa inicial de $\mathbf{X}$ pode aumentar a velocidade de convergência e moldar o resultado da segmentação final. O desenvolvimento de uma estimação para parâmetros γ_k durante as iterações do algoritmo pode ser feita, por exemplo, utilizando-se o número de pixels com o rótulo k durante este processo de iteração. Por fim, a informação das bordas da imagem pode ser acrescentada, de forma que o parâmetro de interação espacial β varie em função da posição do pixel. Isto pode

ser feito calculando-se o valor do gradiente da imagem.

A implementação do método EM/MPM com a otimização apresentada no algoritmo 3 da seção 3.1 se mostrou eficiente. Existe, contudo, a possibilidade da implementação do amostrador de Gibbs de forma paralela, como encontrado no trabalho de Bader *et al.* [BJC95], aumentando a velocidade da segmentação proporcionalmente ao número de processadores da máquina utilizada.

Para imagens com macrotexturas, a abordagem multiresolução deste trabalho, apesar de menos sofisticada que a do modelo autoregressivo proposto por Comer e Delp [CD95, Com95, CD99], foi também satisfatória. Diminuindo a escala da imagem, os detalhes da granulosidade das texturas são suavizados de forma a se obter uma região mais homogênea, facilitando sua localização. Uma desvantagem, entretanto, é o desaparecimento das regiões pequenas mas importantes, sendo necessária uma visão subjetiva na escolha do parâmetro M do número de resoluções.

O uso da técnica de estimação inicial das médias por limiarização, descrita na seção 3.2, teve uma notável eficácia nos resultados das segmentações finais das imagens. Apesar das médias não possuírem exatamente seus valores reais, a estimativa é consideravelmente melhor que a função dada na equação (2.31), induzindo uma convergência mais rápida e melhor da segmentação. O autor sugere ainda a pesquisa de outra abordagem: estimar as médias iniciais com os máximos locais do histograma.

A técnica do *annealing* aplicada ao algoritmo EM/MPM, abordagem proposta na seção 3.3, também foi muito importante na geração de resultados corretos. Como dito anteriormente, nas primeiras iterações as variáveis aleatórias X_s de cada pixel s não se encontram em seus rótulos corretos na grande maioria dos casos. O aumento progressivo da interação espacial β é, portanto, plausível. O autor deste trabalho sugere ainda, a fim de tornar o *annealing* mais eficiente, a pesquisa de uma função não linear para a diminuição da temperatura, ou seja, o aumento de β de forma não constante.

O *annealing* é um método interessante também por facilitar a escolha de parâmetros iniciais. Na grande maioria dos casos, o uso de $\beta_0 = 0$ e de um incremento I bem pequeno é suficiente para se obter um bom resultado. Outra sugestão, com o intuito de diminuir a quantidade de parâmetros a serem previamente definidos, é a pesquisa de um critério de parada das iterações, dispensando a escolha do valor $\mathcal{P}$.

O número de regiões que descrevem objetos dentro de uma imagem é, em muitos casos, um valor subjetivo, que depende do olhar crítico do observador. A validação de classes se torna, portanto, um problema de difícil solução. Os estudos realizados e mostrados no capítulo 4 se mostraram eficientes para uma seleção do valor ótimo do número de rótulos L , posterior ao processo de segmentação.

Por fim, a extensão do método EM/MPM para a segmentação de malhas foi também uma grande contribuição deste trabalho. Os resultados mostrados na seção 5.4 são excelentes quando comparados à técnica implementada por Cuadros-Vargas [CV06], mostrando que a modelagem estocástica por campos aleatórios de Markov é aplicável em muitos casos diferentes.

Referências Bibliográficas

- [ABML89] C. A. Ankenbrandt, B. P. Buckles, F. E. Petry M., and Lybanon. Ocean feature recognition using genetic algorithms with fuzzy fitness functions. *Proc. Third Annual Workshop on Space Operations Automation and Robotics*, pages 679–686, July 1989.
- [ABP90] C. A. Ankenbrandt, B. P. Buckles, and F. E. Petry. Scene recognition using genetic algorithms with semantic nets. *Pattern Recognition Letters*, 11:285–293, 1990.
- [AFDV90] J. Albert, F. Ferri, J. Domingo, and M. Vicens. An approach to natural scene segmentation by means of genetic algorithms with fuzzy data. *Fourth National Symposium in Pattern Recognition and Image Analysis*, pages 97–112, 1990.
- [Aka74] Hirotugu Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.
- [Bal03] André G. R. Balan. Técnicas de segmentação de imagens aéreas para contagem de população de aves. Master’s thesis, ICMC, Universidade de São Paulo, São Carlos, SP, Brasil, Fevereiro 2003.
- [Bes74] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society*, 36(2):192–236, 1974.
- [Bha43] A. Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distributions. *Bulletin of the Calcutta Mathematics Society*, 35:99–110, 1943.
- [Bil98] Jeff A. Bilmes. A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. Technical report, Computer Science Division, Department of Electrical Engineering and Computer Science, U.C. Berkeley, April 1998.
- [BJC95] David A. Bader, Joseph JáJá, and Rama Chellappa. Scalable data parallel algorithms for texture synthesis using Gibbs random fields. *IEEE Transactions on Image Processing*, IP-4:1456–1460, 1995.
- [BP92] J. C. Bezdek and S. K. Pal. Fuzzy models for pattern recognition: Methods that search for structures in data. *IEEE Press*, 1992.

- [Bro66] P. Brodatz. *Textures - A Photographic Album for Artists and Designers*. Reinhold, Dover, New York, 1966.
- [BS94] C. Bouman and M. Shapiro. A multiscale random field model for Bayesian image segmentation. *IP*, 3(2):162–177, March 1994.
- [BTAA05] André G. R. Balan, Agma J. M. Traina, Caetano Traina, and Paulo M. Azevedo. Fractal analysis of image textures for indexing and retrieval by content. pages 581–586. 18th IEEE Symposium on Computer-Based Medical Systems, June 2005.
- [CD95] Mary L. Comer and Edward J. Delp. Multiresolution image segmentation. pages 2415–2418, Detroit, MI, May 1995. IEEE Int. Conf. Acoustic, Speech and Signal Processing.
- [CD99] Mary L. Comer and Edward J. Delp. Segmentation of textured images using a multiresolution Gaussian autoregressive model. *IEEE Transactions on Image Processing*, 8(3):408–420, March 1999.
- [CD00] Mary L. Comer and Edward J. Delp. The EM/MPM algorithm for segmentation of textured images: Analysis and further experimental results. *IEEE Transactions on Image Processing*, 9(10):1731–1744, October 2000.
- [Cer85] V. Cerny. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45(1):41–51, 1985.
- [CG92] George Casella and Edward I. George. Explaining the Gibbs sampler. *The American Statistician*, 46(3):167–174, August 1992.
- [Com95] Mary L. Comer. *Multiresolution Image Processing Techniques with Application in Texture Segmentation and Nonlinear Filtering*. PhD thesis, Purdue University, December 1995.
- [CV06] Alex Jesús Cuadros-Vargas. *Geração de Malhas Volumétricas a Partir de Imagens*. PhD thesis, ICMC, Universidade de São Paulo, São Carlos, SP, Brasil, 2006.
- [DE87] H. Derin and H. Elliott. Modeling and segmentation of noisy and textured images using Gibbs random fields. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 9(1):39–55, 1987.
- [DH73] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley, New York, 1973.
- [DLR77] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39:1–38, 1977.
- [Dou98] Edward R. Dougherty. *Random Processes for Image and Signal Processing*. IEEE Series on Imaging Science and Engineering. SPIE/IEEE, 1998.

- [FPD01] Roger Fjortoft, Wojciech Pieczynski, and Yves Delignon. Generalized mixture estimation and unsupervised classification based on hidden Markov chains and hidden Markov fields. In *Scandinavian Conference on Image Analysis*, pages 733–740, 11–14 June 2001.
- [Fuk90] Keinosuke Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press Professional, Inc., San Diego, CA, USA, 2nd edition, 1990.
- [Gal75] M. M. Galloway. Texture analysis using gray level run lengths. *Computer Graphics and Image Processing*, 4:172–179, 1975.
- [GBS90] M. Gonzalez, D. Benitez, and C. P. Suarez. Segmentation and recognition in visual chromatic spaces. *Cybernetics Systems*, 21:241–247, 1990.
- [GDO85] L. Van Gool, P. Dewaele, and A. Oosterlinck. Texture analysis anno 1983. *Computer Vision, Graphics and Image Processing*, 29:336–357, 1985.
- [GG84] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.
- [GG89] I. Gath and A. B. Gev. Unsupervised optimal fuzzy clustering. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 11(7):773–780, 1989.
- [Gon87] Rafael C. Gonzalez. *Digital Image Processing*. Addison-Wesley, second edition, 1987.
- [GP97] Nathalie Giordana and Wojciech Pieczynski. Estimation of generalized multisensor hidden Markov chains and unsupervised image segmentation. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, (5):465–475, 1997.
- [Har79] R. M. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786–804, 1979.
- [IEE90] IEEE. *IEEE Standard 610.4-1990, Standard Glossary of Image Processing and Pattern Recognition Terminology*. IEEE Press, New York, 1990.
- [Jah97] Bernd Jahne. *Digital Image Processing: Concepts, Algorithms, and Scientific Applications*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1997.
- [Jai89] Anil K. Jain. *Fundamentals of Digital Image Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.
- [Kas80] R. L. Kashyap. Inconsistency of AIC rule for estimating the order of autoregressive models. *IEEE Transactions on Automatic Control*, 25(5):996–998, 1980.
- [KGV83] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.
- [LCC01] Ping-Sung Liao, Tse-Sheng Chen, and Pau-Choo Chung. A fast algorithm for multilevel thresholding. *Journal of Information Science And Engineering*, 17(5):713–727, September 2001.

- [LD89] S. Lakshmanan and H. Derin. Simultaneous parameter estimation and segmentation of Gibbs random fields using simulated annealing. *IEEE Transactions on Pattern Analysis Machine Intelligence*, 11(8):799–813, 1989.
- [LLL00] S. Del Lama, I. Lopes, and M. Del Lama. Genetic variability and level of differentiation in wood stork populations. *IBIS*, November 2000.
- [LMZ98] D. A. Langan, J. W. Modestino, and J. Zhang. Cluster validation for unsupervised stochastic model-based image segmentation. *IEEE Transactions on Image Processing*, 7(2):180–224, February 1998.
- [MMP87] J. Marroquin, S. Mitter, and T. Poggio. Probabilistic solution of ill-posed problems in computational vision. *The American Statistician*, 82:76–89, March 1987.
- [MRR⁺53] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.
- [Ots79] N. Otsu. A threshold selection method from gray-level histogram. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-9(1):62–66, January 1979.
- [Pap92] T. N. Pappas. An adaptive clustering algorithm for image segmentation. *IEEE Transactions on Signal Processing*, 40:901–914, April 1992.
- [PGPB05] Mário Augusto Pazoti, Rogério Eduardo Garcia, José Dalton Cruz Pessoa, and Odemir M. Bruno. Comparison of shape analysis methods in characterization of guignardia citricarpa ascospores. *EJB Electronic Journal of Biotechnology*, 8(3):1–6, 2005.
- [RB93] Todd R. Reed and J. M. Hans Du Buf. A review of recent texture segmentation and feature extraction techniques. *Image Understanding*, 57(3):359–372, 1993.
- [RGR97] J. Rajapakse, J. Giedd, and J. Rapoport. Statistical approach to segmentation of single-channel cerebral mr images. *IEEE Transactions on Medical Imaging*, 16(2), April 1997.
- [Ris78] J. J. Rissanen. Modelling by shortest data description. *Automatica*, 14:465–471, 1978.
- [Ron05] Marco Aurélio Roncatti. Desenvolvimento de uma interface para processamento de imagens. Monografia de graduação, Inst. de Ciências Matemáticas e da Computação, Univ. de São Paulo, 2005.
- [Ros97] Anders Rosholm. *Statistical Methods for Segmentation and Classification of Images*. PhD thesis, Technical University of Denmark, 1997.
- [Sch78] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.
- [Scl87] Stanley L. Sclove. Application of model-selection criteria to some problems in multivariate analysis. *Psychometrika*, 52:333–343, 1987.

- [She90] Y. Shen. Cluster validation with applications to image segmentation. Master's thesis, Elect. Eng. Comput. Sci. Dept., Wisconsin University, Milwaukee, WI, 1990.
- [Son99] Millan Sonka. *Image Processing, Analysis and Machine Vision*. PWS, 1999.
- [SP05] W. R. Schwartz and H. Pedrini. Segmentação de imagens de terrenos baseada na associação de características de texturas com dependência espacial modelada por campo aleatório de Markov. Goiânia, 2005. XII Simpósio Brasileiro de Sensoriamento Remoto.
- [ST06] Surgical-Tutor. A free online surgical resource, 2006. Site visitado em março de 2006. <http://www.surgical-tutor.org.uk/>.
- [WS90] R. Wilson and M. Spann. A new approach to clustering. *Pattern Recognition*, 23(12):1413–1425, 1990.
- [Zha88] J. Zhang. *Two-Dimensional Stochastic Model-Based Image Analysis*. PhD thesis, Rensselaer Polytechnic Institute, Troy, NY, August 1988.
- [ZML94] J. Zhang, J. W. Modestino, and D. A. Langan. Maximum-likelihood parameter estimation for unsupervised model-based image segmentation. *IEEE Transactions on Image Processing*, 3(4):404–420, July 1994.