

Virgínia Fernandes Mota

Segmentação de Múltiplos Objetos em Imagens

Orientador:
Marcelo Bernardes Vieira

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Juiz de Fora
Dezembro, 2008

Monografia submetida ao corpo docente do Instituto de Ciências Exatas da Universidade Federal de Juiz de Fora como parte integrante dos requisitos necessários para obtenção do grau de bacharel em Ciência da Computação

Prof. Marcelo Bernardes Vieira, D. Sc.
Orientador

Prof. Marcelo Lobosco, D. Sc.

Prof. Wilhelm Passarella Freire, D. Sc.

Sumário

Lista de Figuras

Resumo

1	Introdução	p. 7
1.1	Objetivos	p. 8
1.2	Estrutura do Trabalho	p. 8
2	Fundamentos	p. 9
2.1	Função característica	p. 9
2.2	Otimização	p. 10
2.2.1	Segmentação como um problema de otimização	p. 10
2.3	Modelo de Gibbs	p. 10
2.4	Multiresolução	p. 11
2.5	Técnicas de Otimização	p. 14
2.5.1	Corte em Grafos	p. 14
2.5.2	Otimização Utilizando Multiresolução	p. 15
2.6	Modelo de Misturas Gaussianas	p. 16
2.6.1	O método <i>Expectation-Maximization</i>	p. 17
3	Modelo Computacional	p. 19
3.1	Visão Geral	p. 19
3.1.1	Cálculo das Cores mais Significativas	p. 19

3.1.2	Criação da Energia	p. 21
3.1.3	Montagem do Grafo e Minimização da Energia	p. 22
3.2	Resultados	p. 22
4	Conclusão	p. 27
	Referências	p. 29

Lista de Figuras

1	Exemplo de segmentação	p. 7
2	Exemplo de segmentação construindo a função característica	p. 9
3	Fator de escala de uma função <i>wavelet</i> Ψ (REIGOTA, 2007)	p. 13
4	Fator de deslocamento: à direita a função <i>wavelet</i> $\Psi(t)$, à esquerda a função <i>wavelet</i> deslocada $\Psi(t - b)$	p. 13
5	Representação piramidal de uma imagem com 3 níveis de resolução	p. 15
6	Pixel na escala j representa 4 pixels na escala $j-1$	p. 16
7	Diagrama do funcionamento geral do segmentador	p. 19
8	Exemplo de marcação de cores	p. 20
9	Imagem original e imagem do gradiente de Di Zenzo	p. 21
10	Segmentação de um objeto utilizando $ng_o = 6$ e $ng_f = 2$. A imagem possui 800x600 pixels e 24 bits.	p. 23
11	Segmentação de nove objetos utilizando $ng_o = 6$ e $ng_f = 2$. A imagem possui 636x876 pixels e 24 bits.	p. 23
12	Segmentação de um objeto utilizando $ng_o = 6$ e $ng_f = 2$. A imagem possui 800x549 pixels e 24 bits.	p. 24
13	Segmentação de um objeto utilizando $ng_o = 6$ e $ng_f = 2$. A imagem possui 548x510 pixels e 24 bits.	p. 24
14	Segmentação de um objeto utilizando $ng_o = 10$ e $ng_f = 6$. A imagem possui 548x510 pixels e 24 bits.	p. 25
15	Segmentação de três objetos utilizando $ng_o = 6$ e $ng_f = 2$. A imagem possui 800x600 pixels e 24 bits.	p. 25

Resumo

A segmentação está presente em muitas aplicações nas áreas de processamento de imagens, visão computacional e computação gráfica. Desta forma, uma segmentação satisfatória pode ser crucial para o sucesso da aplicação. Uma das técnicas mais difundidas, e a utilizada neste trabalho, é tratar a segmentação como um problema de otimização. O objetivo dessa monografia foi implementar um segmentador para múltiplos objetos em imagens com fundo com pouca variação de cor. A técnica de misturas de gaussianas se mostrou muito adequada ao problema, o que foi provado através de vários experimentos, proporcionando, assim, resultados satisfatórios.

1 *Introdução*

A segmentação é, geralmente, a primeira etapa realizada em diversas soluções nas áreas de processamento de imagens, visão computacional e computação gráfica, como na reconstrução automática de documentos fragmentados e na localização de determinado objeto em uma cena. Se essa tarefa for mal realizada, acarretará na má realização das tarefas restantes.

A idéia é determinar, em uma dada imagem, o que é considerado objeto e o que é considerado fundo. A Figura 1 ilustra uma imagem digital e a segmentação de um objeto desta imagem.

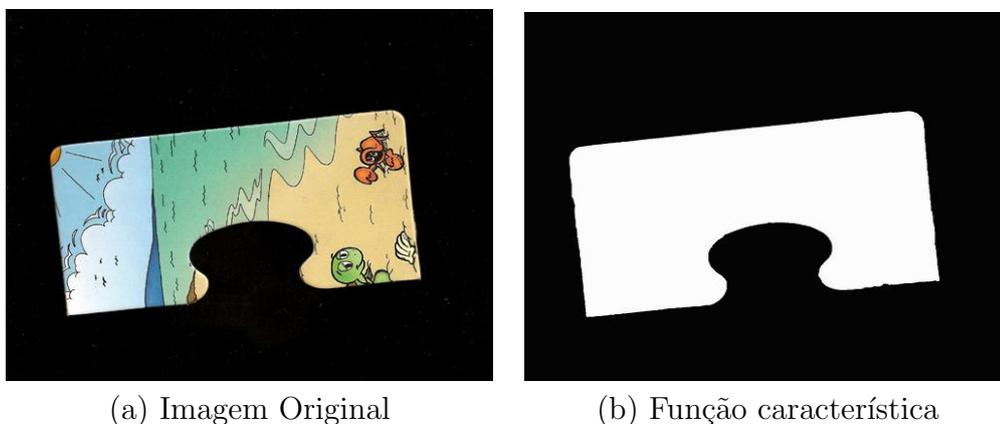


Figura 1: Exemplo de segmentação

O problema tratado nessa monografia é o de segmentação de múltiplos objetos arbitrários em uma imagem com fundo com pouca variação de cor. Ou seja, obter uma função característica que representa fielmente o suporte dos objetos, preservando seus contornos.

1.1 Objetivos

O objetivo primário deste trabalho é estudar, desenvolver e implementar métodos de segmentação para o problema de se obter múltiplos objetos em um fundo com pouca variação de cor.

Como consequência do objetivo primário deseja-se:

- criar um método confiável e robusto, isto é, que consiga segmentar os objetos de qualquer imagem de forma correta;
- estudar formas para obtenção das cores mais significativas da imagem;
- aplicar a segmentação em um problema específico de reconstrução automática de quebra-cabeça.

1.2 Estrutura do Trabalho

Este trabalho apresenta alguns métodos para resolução do problema segmentação de imagens, tratando-a como um problema de otimização (Seções 2.2 e 2.5.1) e utilizando a multiresolução para sua realização, através da teoria de *Wavelets* (Seções 2.4 e 2.5.2).

Para um bom entendimento do processo de segmentação, as seções indicadas são: 2.1, 2.2, 3.1.1 e 3.1.2.

Se o interesse for técnicas de minimização de energia, as seções 2.2.1, 2.3, 2.5.1 e 3.1.3 apresentam explicações claras sobre a energia de Gibbs, minimização via corte em grafos e otimização.

Para as técnicas de cálculo das cores significativas da imagem, etapa inicial do processo de segmentação, as seções adequadas são 2.6 e 3.1.1, onde são tratadas a construção de histogramas e o uso de *Gaussian Mixture Models*.

Uma visão geral do segmentador implementado neste trabalho é mostrada em 3.1.

Os resultados obtidos através do método criado podem ser vistos em 3.2.

2 Fundamentos

Este capítulo apresentará as noções matemáticas que caracterizam o problema e viabilizam a sua solução.

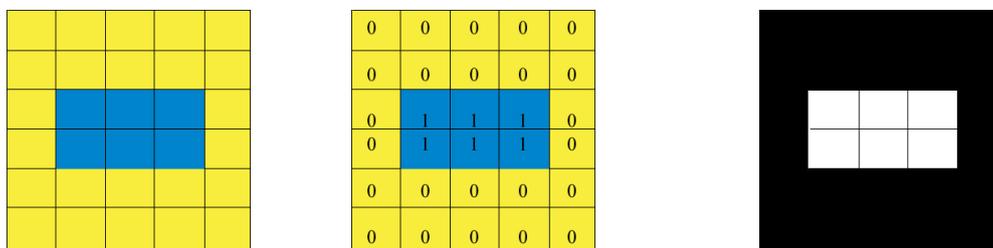
2.1 Função característica

As técnicas de segmentação subdividem a imagem em regiões baseadas em um critério específico. No caso deste trabalho, a subdivisão será em fundo e objeto. Isso será conseguido através da construção da função característica da imagem.

A função característica $f(x)$ carrega toda a informação da segmentação e é definida por

$$f(x) = \begin{cases} 1, & x \in \text{Objeto} \\ 0, & x \in \text{Fundo} \end{cases} \quad (2.1)$$

O problema da segmentação pode ser tratado como encontrar a função característica da imagem, tratando a imagem como uma matriz onde cada pixel é um elemento dessa e rotulando cada pixel como sendo do fundo (0) ou do objeto (1), como mostra a Figura 2. Essa segmentação é chamada segmentação rígida.



(a) Imagem Original (b) Função característica (c) Resultado da segmentação

Figura 2: Exemplo de segmentação construindo a função característica

2.2 Otimização

Otimização se refere ao estudo de problemas nos quais se deseja minimizar ou maximizar uma função real através de escolhas sistemáticas de valores de variáveis reais ou inteiras dentre um conjunto permitido, ou seja, dada uma função $f : A \rightarrow \mathfrak{R}$ de algum conjunto A para os números reais, deseja-se obter um elemento x_0 em A tal que $f(x_0) \leq f(x), \forall x \in A$ (minimização) ou tal que $f(x_0) \geq f(x), \forall x \in A$ (maximização).

Normalmente, A é um subconjunto do espaço euclidiano \mathfrak{R}^n com algumas restrições que seus elementos devem obedecer. Tal subconjunto recebe o nome de espaço de busca, seus elementos são chamados de soluções possíveis, ou soluções candidatas, e a função f de função objetivo. Uma solução possível que maximiza, ou minimiza, dependendo do objetivo, a função objetivo é chamada de solução ótima.

2.2.1 Segmentação como um problema de otimização

É possível modelar o problema de segmentação de forma que se consiga a função característica através da minimização de uma função objetivo. Ou seja, dado um conjunto matricial M , deve-se encontrar a função característica X que seja o argumento mínimo de uma função objetivo f :

$$\operatorname{argmin}_M f(X) \quad (2.2)$$

Um tipo de função objetivo muito utilizada na área de segmentação de imagens é a energia de Gibbs, como pode ser visto em (LI et al., 2004) e (ROTHER; KOLMOGOROV; BLAKE, 2004).

2.3 Modelo de Gibbs

A energia de Gibbs é definida da seguinte forma:

$$E(X) = \sum_{x_i \in V} E_1(X(x_i)) + \lambda \sum_{x_i, x_j \in \epsilon} E_2(X(x_i), X(x_j)) \quad (2.3)$$

onde x_i e x_j são elementos do conjunto a ser segmentado, V é o conjunto de elementos, ϵ o conjunto dos elementos com relação de vizinhança entre si e λ é uma constante com o objetivo de dar uma importância maior para um dos termos.

E_1 é o termo que define um custo para cada x_i pertencer a um dos conjuntos. Com

o objetivo de minimizar a função objetivo, tal custo deve ser inversamente proporcional à probabilidade de x_i pertencer ao conjunto. Geralmente é dado na forma:

$$\begin{cases} E_1(X(x_i) = 1) = 0 & E_1(X(x_i) = 0) = \infty & \forall x_i \in O \\ E_1(X(x_i) = 1) = \infty & E_1(X(x_i) = 0) = 0 & \forall x_i \in F \\ E_1(X(x_i) = 1) = \phi(\rho_o) & E_1(X(x_i) = 0) = \phi(\rho_f) & \forall x_i \in N \end{cases} \quad (2.4)$$

onde O são os elementos x_i que já tem rótulo de objeto, F os elementos que já tem rótulo de fundo, N os elementos que ainda não estão rotulados e ϕ uma função inversamente proporcional ao seu parâmetro.

E_2 é o termo que define uma penalidade para que dois elementos vizinhos pertençam a conjuntos diferentes. Esta penalidade deve ser tal que mantenha as discontinuidades originais. Ou seja, o custo que E_2 atribui para dois elementos x_i e x_j depende da semelhança entre eles. Se a semelhança for grande, significa que eles têm maior probabilidade de estarem no mesmo conjunto, tornando o custo maior. Caso contrário, o custo diminui.

2.4 Multiresolução

Algumas das técnicas para segmentação como problema de otimização utilizam a multiresolução.

Uma representação diádica em multiresolução do espaço $L^2(\mathfrak{R})$ consiste na seqüência de sub-espacos aninhados

$$\dots \subset V_0 \subset V_1 \subset \dots \subset V_n \subset V_{n+1} \subset \dots \subset L^2(\mathfrak{R})$$

que satisfaz as seguintes relações:

- $\cap_j V_j \neq \{\}, \cup_j V_j = L^2(\mathfrak{R})$
- $f(t) \in V_j \Leftrightarrow f(2t) \in V_{j+1}$
- $f(t) \in V_0 \Rightarrow f(t - k) \in V_0$
- Existe uma função $\phi(t)$, chamada função de escalamento, tal que $\{\phi(t - k)\}$ é uma base ortonormal de V_0

Uma decomposição em multiresolução é obtida pela Transformada de *Wavelets*. As *Wavelets* são aproximações que usam superposição de funções. Existem desde o início do

século XIX, quando Joseph Fourier descobriu ser possível representar funções periódicas através de funções seno e cosseno.

A Transformada de *Wavelets* pode ser vista como técnica por janelas, com dimensão variável da janela, o que permite avaliação do sinal sob análise, no espaço-tempo \times frequência, e a identificação de quais componentes espectrais existem em um dado intervalo de tempo. Essa característica da transformada torna-a mais apropriada também para aproximar funções com descontinuidades.

Há duas versões da transformada de *wavelets*: a contínua e a discreta. A contínua, análoga à Transformada de Fourier, é usada principalmente em análise e caracterização de sinais. A discreta, análoga à Transformada Discreta de Fourier, é a mais apropriada para compressão de dados e reconstrução de sinais.

A transformada contínua de *Wavelets* é a soma, ao longo do tempo, do sinal, multiplicado pela *wavelet* em certa escala e deslocada de certa quantidade. Este processo produz coeficientes *wavelet* que são funções da escala e da posição.

Matematicamente, pode-se definir a Transformada Contínua de *Wavelets* em $F(a, b)$ como:

$$F(a, b) = \int_{-\infty}^{\infty} f(t)\Psi_{a,b}(t)dt \quad (2.5)$$

As variáveis a e b são valores reais, em que a é um parâmetro de escala (contração ou dilatação) e b é um parâmetro de localização (deslocamento). A função $\Psi_{a,b}(t)$ é denominada *wavelet* e é definida como:

$$\Psi_{a,b}(t) = \frac{1}{\sqrt{a}}\Psi\left(\frac{t-b}{a}\right), a \neq 0, b \in \mathfrak{R} \quad (2.6)$$

A *wavelet* mãe precisa satisfazer a seguinte propriedade:

$$\int \Psi(t)dt = 0 \quad (2.7)$$

O efeito do parâmetro a na função *wavelet* pode ser observado na Figura 3. Quanto menor a escala, mais comprimida será a função *wavelet*, e vice-versa. Então, existe uma relação entre a escala e a frequência revelada pela análise por *wavelet*:

- menor escala $a \Rightarrow$ *wavelet* comprimida \Rightarrow detalhes mudando rapidamente \Rightarrow alta frequência w
- maior escala $a \Rightarrow$ *wavelet* dilatada \Rightarrow características globais mudando lentamente

\Rightarrow baixa frequência w .

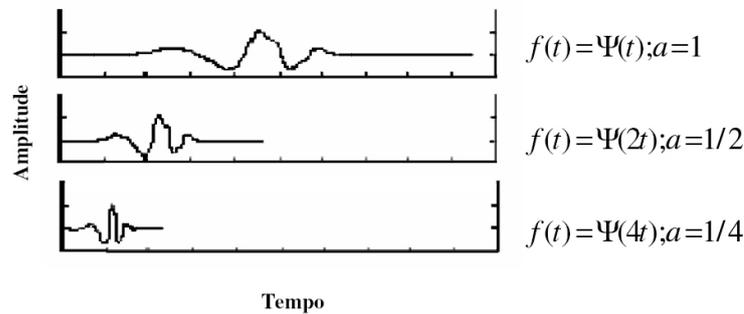


Figura 3: Fator de escala de uma função *wavelet* Ψ (REIGOTA, 2007)

Outro fator presente na análise por *wavelet* é o fator de deslocamento (b), representado por $f(t - b)$. Pode-se observar o efeito de deslocamento na Figura 4.

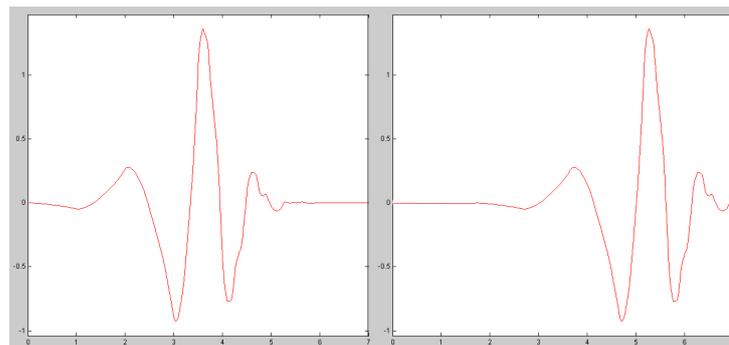


Figura 4: Fator de deslocamento: à direita a função *wavelet* $\Psi(t)$, à esquerda a função *wavelet* deslocada $\Psi(t - b)$.

A Transformada de *Wavelet* pode ser tratada sob sua forma discreta, sem prejuízo de suas qualidades e propriedades, e de maneira numericamente estável (REIGOTA, 2007).

Na Transformada Discreta de *Wavelet* o parâmetro de escala a e de deslocamento b assumem apenas valores discretos. Para a escolhe-se valores inteiros, potências de um parâmetro fixo a_0 , sendo que geralmente a_0 é uma potência de 2:

$$a = a_0^{-j} \quad a_0 > 1 \quad e \quad j \in \mathbb{Z} \quad (2.8)$$

Diferentes valores de j correspondem a *wavelets* de diferentes larguras.

A discretização do parâmetro b deve depender de j : *wavelets* estreitas (alta frequência) são transladadas por pequenas distâncias a fim de cobrir todo o domínio do tempo, enquanto que *wavelets* mais largas (frequência mais baixa), devem ser transladadas por uma

distância maior. Assim, a escolha natural é:

$$b = kb_0 a_0^{-j} \quad b_0 > 0, \text{ fixo}, j, k \in Z \quad (2.9)$$

A *wavelet* discreta fica então:

$$\Psi_{j,k}(t) = a_0^{\frac{j}{2}} \Psi(a_0^j t - kb_0) \quad (2.10)$$

A Transformada de *Wavelets* para sinais discretos é então definida como:

$$F_{m,k}(a, b) = a_0^{\frac{-j}{2}} \sum f(t) \Psi(a_0^j t - kb_0), \quad a \text{ e } b \in Z \quad (2.11)$$

2.5 Técnicas de Otimização

Nesta seção, todos os elementos matemáticos apresentados antes são combinados para que seja possível modelar o problema de segmentação na forma de um problema de corte em grafos e de um problema de otimização utilizando multiresolução.

2.5.1 Corte em Grafos

Como foi mencionado na seção 2.2.1 os problemas de segmentação de imagens podem ser facilmente definidos na forma de minimização de energia. Porém, a tarefa computacional de minimizar energias costuma ser muito custosa.

Uma forma de se abordar a segmentação por otimização é transformá-la em um problema de corte em grafos (RODRIGUES, 2007). A idéia básica é construir um grafo especializado para a função de energia ser minimizada, tal que o corte mínimo no grafo minimize também a energia.

Um grafo é um par $G = \langle V, \epsilon \rangle$ de conjuntos tais que $\epsilon \subseteq [V]^2$. Os elementos do conjunto V são os vértices ou nós do grafo G , e os do conjunto ϵ suas arestas.

Um grafo pode ser orientado, ou seja, suas arestas contém informação de direção, e assim, são chamadas de arcos. Neste caso, uma aresta dada por (x_i, x_j) representa um arco que vai do vértice x_i para o vértice x_j . Isso não implica que haja um arco inverso. Ele pode conter também pesos, ou custos, para cada aresta. O peso de uma aresta $e = (x, y)$ é denotado por $c(x, y)$.

Supondo um grafo direcionado $G = \langle V, \epsilon \rangle$ com dois vértices especiais, ou terminais,

s e t , um corte- s - t $C = \epsilon(S, T)$ é uma partição dos vértices em V em dois conjuntos separados S e T , tais que $s \in S$ e $t \in T$. O problema do corte mínimo é achar um corte C com o menor custo.

Um corte $C = \epsilon(S, T)$ pode ser denotado como uma função característica $f(x_i)$ para todo vértice $x_i \in V - \{s, t\}$, onde:

$$f(x) = \begin{cases} 0, & x_i \in S \\ 1, & x_i \in T \end{cases} \quad (2.12)$$

Cada corte em G tem um custo, logo, G representa o mapeamento de função de energia de todos os cortes no grafo G para o conjunto dos números reais não negativos. Portanto, a energia E que G representa pode ser vista como uma função de n variáveis binárias: $E(f(x_i)), \forall x_i \in V$ é igual ao custo do corte definido por 2.12.

Uma função E de n variáveis binárias pode ser representada por um grafo se existe um grafo $G = \langle V, \epsilon \rangle$ com terminais s e t e um subconjunto de vértices $V_0 = x_0, \dots, x_n \subset V - \{s, t\}$ tal que para qualquer configuração $f(x_i)$ o valor da energia $E(f(x_i))$ é igual a uma constante mais um custo de um corte- s - t mínimo entre todos os cortes $C = \epsilon(S, T)$, nos quais $x_i \in S$ se $f(x_i) = 0$, e $x_i \in T$ se $f(x_i) = 1$, $\forall_i \in 0, \dots, n$. É dito que E é exatamente representado por G se essa constante for zero.

2.5.2 Otimização Utilizando Multiresolução

A idéia por trás das técnicas de otimização utilizando multiresolução é que ao diminuir a resolução, o ruído e as complexidades das texturas são reduzidos, porque o nível de detalhe está diminuindo. Assim, os objetos na imagem serão mais significativos, já que tem-se menos altas frequências que poderiam atrapalhar a localização dos objetos (BOCCIGNONE et al., 2007).

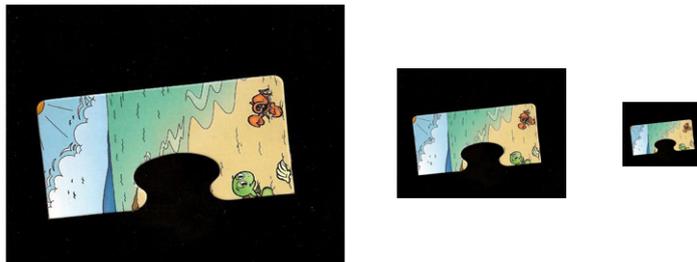


Figura 5: Representação piramidal de uma imagem com 3 níveis de resolução

Em (KOEPLER; LOPEZ; MOREL, 1994), a segmentação é tratada como o encontro de várias regiões na imagem e não é única. Na verdade, tem-se uma hierarquia de segmentações de uma escala mais fina até uma mais grosseira.

Em (CHOI; BARANIUK, 2001) as informações são passadas de pai para filho, uma vez que um pixel em uma escala maior é pai de quatro pixels de uma escala menor. Assim, tem-se aqui também um algoritmo de segmentação passando em várias escalas e propagando as informações para a próxima.

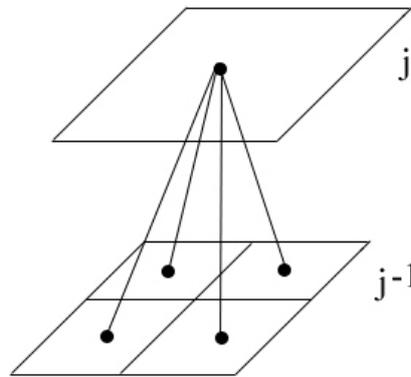


Figura 6: Pixel na escala j representa 4 pixels na escala $j-1$

É possível perceber que as técnicas citadas utilizam apenas o espaço de escala. De fato, geralmente é este o mais utilizado, porém o espaço de detalhes acrescenta informação importante sobre as altas frequências, como por exemplo, as bordas dos objetos ficam mais definidas neste espaço.

2.6 Modelo de Misturas Gaussianas

A maioria das técnicas de segmentação trabalha com as cores mais significativas da imagem para encontrar fundo e objeto. O método mais simples é montar o histograma da imagem, no qual cada cor presente na imagem é contada. As cores mais significativas, então, são aquelas que aparecem mais vezes na imagem.

Métodos baseados em histograma são muito limitados, logo se faz necessário um método mais sofisticado que seja capaz de, com poucas amostras, cobrir satisfatoriamente o universo de amostras.

O modelo de misturas gaussianas (ou *Gaussian Mixture Models* ou GMM) formam

aglomerados representando a função de probabilidade da densidade de variáveis observadas como uma mistura de gaussianas multivariadas. Este pode ser aplicado como modelo de cores de um objeto para resolver tarefas como segmentação e localização de objetos em tempo real baseado em cores (RAJA; MCKENNA; GONG, 1998).

A densidade condicional para um pixel, ξ , pertencer a um objeto, O , é modelada como a mistura com m componentes de densidade:

$$p(\xi|O) = \sum_{j=1}^m p(\xi|j)P(j) \quad (2.13)$$

onde $P(j)$ corresponde à probabilidade do pixel ξ ser gerado pelo componente j e

$$\sum_{j=1}^m P(j) = 1 \quad (2.14)$$

Cada componente do modelo de mistura é uma Gaussiana com média μ e matriz de covariância S . Em um espaço de cores 2D tem-se:

$$p(\xi|j) = \frac{1}{2\pi|\Sigma_j|^{\frac{1}{2}}} e^{-\frac{1}{2}(\xi-\mu_j)^T \Sigma_j^{-1}(\xi-\mu_j)} \quad (2.15)$$

O problema está em descobrir as componentes gaussianas da GMM que descrevem a distribuição de cores da imagem de forma satisfatória. Assim, é necessário estimar a média e a covariância da gaussiana. Um dos métodos mais utilizados para o encontro destes valores é o *Expectation-Maximization*.

2.6.1 O método *Expectation-Maximization*

O método *Expectation-Maximization*(EM) é usado para achar a estimativa de parâmetros que estão faltando ou estão incompletos. No caso da segmentação, deseja-se saber quais gaussianas se ajustam às cores da imagem calculando suas médias e covariâncias.

O EM é formado por duas etapas: a etapa de estimação, na qual estima-se os valores que faltam (média e covariância, no caso); e a etapa de maximização, na qual maximiza os valores para se ajustar aos dados. Essas duas etapas são repetidas até que um critério de parada seja satisfeito (CARSON et al., 2002).

Sendo K o número de gaussianas no modelo de misturas, a densidade de probabilidade

é dada por:

$$f(x|\Theta) = \sum_{i=1}^K \alpha_i f_i(x|\theta_i) \quad (2.16)$$

onde x é um vetor de cores, α_k são pesos, θ a coleção de parâmetros e f_i uma gaussiana multivariada com média μ e covariância Σ na forma:

$$f_i(x|\Theta_i) = \frac{1}{2\pi|\Sigma_j|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_j)^T \Sigma_j^{-1} (x-\mu_j)} \quad (2.17)$$

As equações de atualização são dadas na seguinte forma:

$$\alpha_i^{new} = \frac{1}{N} \sum_{j=1}^N p(i|x_j, \Theta^{old}) \quad (2.18)$$

$$\mu_i^{new} = \frac{\sum_{j=1}^N x_j p(i|x_j, \Theta^{old})}{\sum_{j=1}^N p(i|x_j, \Theta^{old})} \quad (2.19)$$

$$\sum_i^{new} = \frac{\sum_{j=1}^N p(i|x_j, \Theta^{old}) (x_j - \mu_i^{new})(x_j - \mu_i^{new})^T}{\sum_{j=1}^N p(i|x_j, \Theta^{old})} \quad (2.20)$$

onde N é o número de pixels e $p(i|x_j, \Theta)$ é a probabilidade da gaussiana i se ajustar ao pixel x_j dado o dado Θ :

$$p(i|x_j, \Theta) = \frac{\alpha_i f_i(x_j|\theta_i)}{\sum_{k=1}^K \alpha_k f_k(x_j|\theta_k)} \quad (2.21)$$

Essas atualizações garantem o encontro de todas as matrizes de covariância e as restringe a matrizes diagonais ou a constante vezes a matriz identidade.

As atualizações são repetidas até

$$\log L(\Theta|X) = \log \prod_{k=1}^N f(x_k|\Theta) \quad (2.22)$$

crescer menos de 1% entre uma iteração e outra (se isso não acontecer em 10 iterações, a parada é forçada). O número de iterações não precisa ser fixo, caso se deseje refinar a solução, basta aumentar o número de iterações.

3 Modelo Computacional

Este capítulo apresentará as soluções adotadas para resolver o problema de segmentação.

3.1 Visão Geral

A Figura 7 mostra um diagrama do funcionamento geral do segmentador implementado neste trabalho.

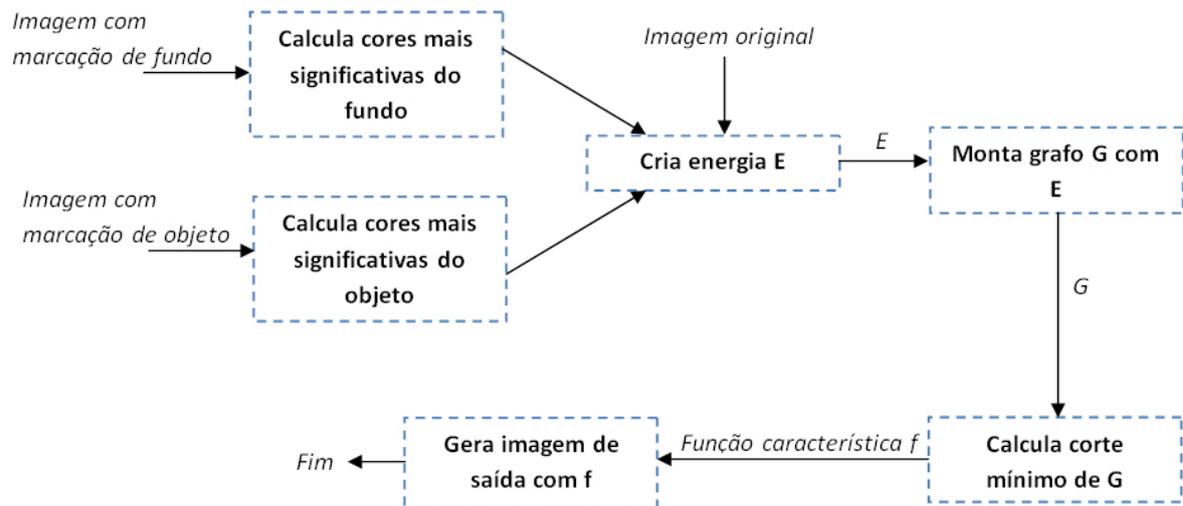


Figura 7: Diagrama do funcionamento geral do segmentador

3.1.1 Cálculo das Cores mais Significativas

Para iniciar a segmentação é necessário informações sobre o que é considerado fundo e objeto. Para tal, basta fazer pequenas marcações na imagem original, como mostra a Figura 8.



Figura 8: Exemplo de marcação de cores

Uma das formas para calcular as cores mais significativas é montando o histograma, um para o fundo e um para objeto, com as cores marcadas. Depois basta ordená-lo e escolher as k primeiras cores. Mas esta abordagem não traduz a distribuição de probabilidade satisfatoriamente e k não é simples de ser calculado, pois não pode ser muito pequeno, para não perder informação, e nem muito grande, para não perder eficiência.

Outra forma é transformar a imagem para o espaço *Lab*, que por ser uniforme, torna as distâncias entre as cores significativas (CARSON et al., 2002). Mas esta transformação é muito custosa, portanto, não é eficiente.

Neste trabalho, a partir das cores marcadas, são calculadas as cores mais significativas para o fundo e para o objeto usando modelos de misturas gaussianas. Foram usadas duas GMM: uma para o fundo e uma para o objeto. Para o fundo foram escolhidas menos gaussianas, porque este trabalho tem como objetivo segmentar imagens com fundo com pouca variação de cor.

A primeira etapa para o cálculo das cores mais significativas é montar um histograma de cores, um para o fundo e um para o objeto, em ordem decrescente. As cores do histograma serão as amostras para a criação das componentes gaussianas das GMM.

Para estimar as GMM foi utilizado o método *Expectation-Maximization*. Como este trabalho utiliza imagens coloridas, cada pixel possui como cor um vetor de cores RGB, assim o pixel x e a média μ das Equações 2.17, 2.18, 2.19, 2.20 e 2.21 são vetores de três posições.

Para evitar que as matrizes de covariância sejam singulares ou sejam mal condicionadas, por causa dos valores muito próximos de zero que o algoritmo calcula, a cada iteração do algoritmo é adicionado um ruído à diagonal da matriz com distribuição uniforme entre 0 e 1.

3.1.2 Criação da Energia

Neste trabalho, a segmentação foi tratada como um problema de minimização de energia. Essa energia segue o modelo de Gibbs mostrado na Equação 2.3.

O primeiro termo, E_1 , levará em consideração a probabilidade do pixel pertencer à GMM do fundo ou do objeto.

$$\begin{cases} E_1(X(x_i) = 1) = 0 & E_1(X(x_i) = 0) = \infty & \forall x_i \in O \\ E_1(X(x_i) = 1) = \infty & E_1(X(x_i) = 0) = 0 & \forall x_i \in F \\ E_1(X(x_i) = 1) = P_F/(P_O + P_F) & E_1(X(x_i) = 0) = P_O/(P_O + P_F) & \forall x_i \in N \end{cases} \quad (3.1)$$

onde

$$\begin{aligned} P_O &= \sum_{i=1}^{ng_o} f_i^O \alpha_i^O \\ P_F &= \sum_{i=1}^{ng_f} f_i^F \alpha_i^F \end{aligned} \quad (3.2)$$

sendo ng_o o número de gaussianas da GMM do objeto, ng_f o número de gaussianas da GMM do fundo, f_i^F e f_i^O as componentes gaussianas do fundo e do objeto, respectivamente, e α_i^O e α_i^F os pesos de cada gaussiana onde $\sum_i \alpha_i = 1$. P_F e P_O estão normalizados entre seus valores mínimos e máximos.

O segundo termo, E_2 , é definido como uma função do gradiente de cor entre dois pixels x_i e x_j :

$$E_2(X(x_i, x_j)) = g(C_{x_i, x_j}) \quad (3.3)$$

com $g(\xi) = \frac{1}{\xi+1}$ e $C_{x_i, x_j} = \|C(x_i) - C(x_j)\|^2 * (\nabla_{DZ}(x_i) + \nabla_{DZ}(x_j))$, que é a norma-L2 da diferença de cores RGB dos pixels x_i e x_j multiplicada pelo gradiente de Di Zenzo (∇_{DZ}) de cada um dos pixels.

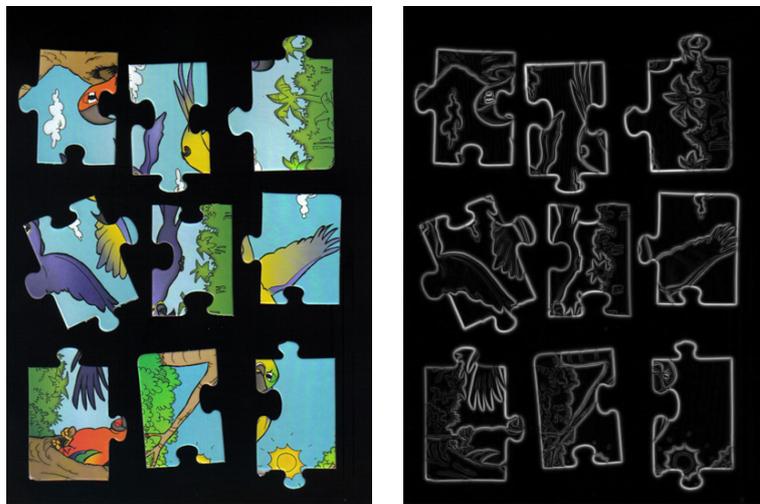


Figura 9: Imagem original e imagem do gradiente de Di Zenzo

O gradiente de Di Zenzo (ZENZO, 1986) é calculado a partir das derivadas em x e y , de cada um dos canais RGB, obtidas através do método de Shen-Castan (SHEN; CASTAN, 1987), que é um operador derivativo ótimo para encontrar múltiplas bordas de passo. A imagem resultante, mostrada na Figura 9, é utilizada para armazenar o valor do gradiente de cada um dos pixels. Nota-se que a imagem contém apenas as altas frequências da imagem, por isso ela foi utilizada na E_2 , para aumentar a energia das bordas.

3.1.3 Montagem do Grafo e Minimização da Energia

Para a minimização da energia, foi montado um grafo com vizinhança 8-conexa, isto é, cada vértice é conectado aos 8 vizinhos mais próximos, a partir da energia mostrada em 3.1.2 e da imagem original da seguinte forma:

- Cada pixel x_i representa um vértice do grafo G ;
- Cada pixel x_i possui custo equivalente à $E_1(x_i)$;
- A aresta (x_i, x_j) de G tem custo equivalente à $E_2(x_i, x_j)$.

Partindo do fato de que os métodos de corte em grafos acham qualquer corte presente no grafo, é possível afirmar que um algoritmo de segmentação que use corte em grafos é capaz de localizar mais de um objeto na imagem, como pode ser visto na Figura 10. O algoritmo escolhido para o cálculo do corte mínimo foi o proposto em (KOLMOGOROV; ZABIN, 2004).

3.2 Resultados

Nesta seção encontram-se alguns resultados obtidos com a aplicação desenvolvida neste trabalho. Inicialmente, o número de componentes gaussianas para cada GMM (ng_o para objeto e ng_f para o fundo) se manteve fixo, mas foi possível perceber que aumentando o número de gaussianas, a qualidade de segmentação melhora e possibilita a segmentação de imagem mais complexas. Isto pode ser observado nas Figuras 13 e 14.

As figuras a seguir mostram a imagem original, a imagem de marcação de fundo, a imagem de marcação de objeto, o gradiente de Di Zenzo, usando o operador ótimo para derivada parcial de Shen-Castan, a função característica e a segmentação obtida.

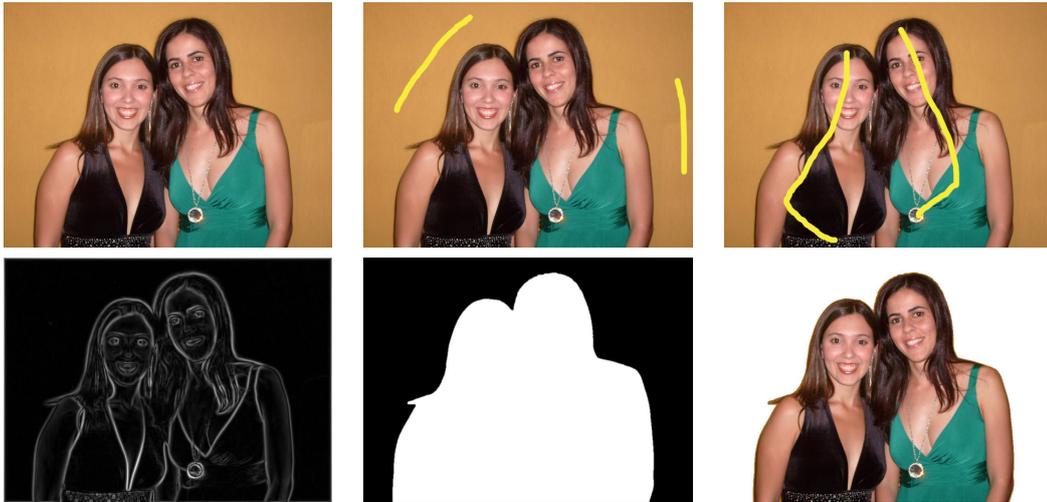


Figura 10: Segmentação de um objeto utilizando $ng_o = 6$ e $ng_f = 2$. A imagem possui 800x600 pixels e 24 bits.

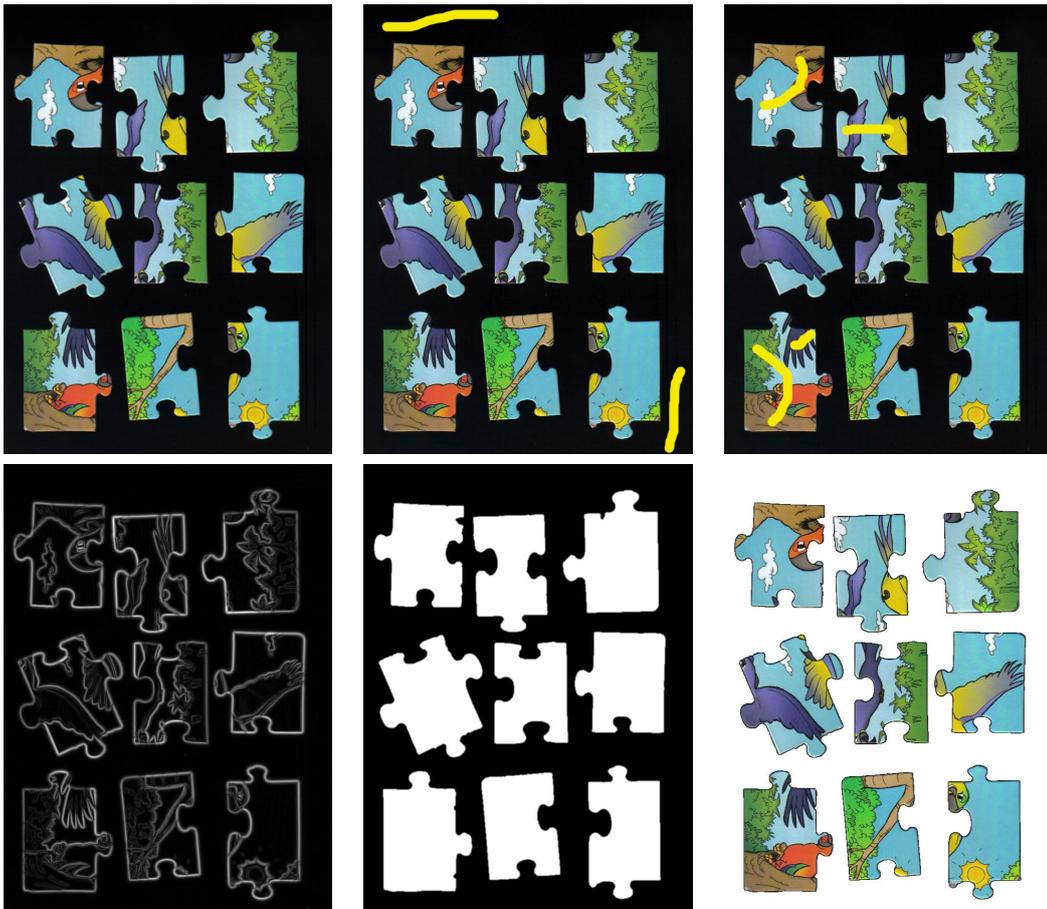


Figura 11: Segmentação de nove objetos utilizando $ng_o = 6$ e $ng_f = 2$. A imagem possui 636x876 pixels e 24 bits.

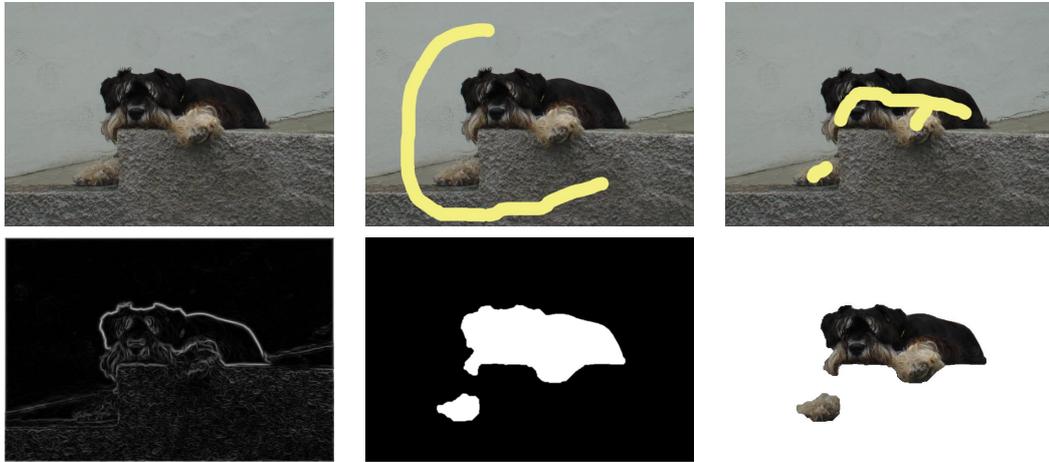


Figura 12: Segmentação de um objeto utilizando $ng_o = 6$ e $ng_f = 2$. A imagem possui 800x549 pixels e 24 bits.



Figura 13: Segmentação de um objeto utilizando $ng_o = 6$ e $ng_f = 2$. A imagem possui 548x510 pixels e 24 bits.

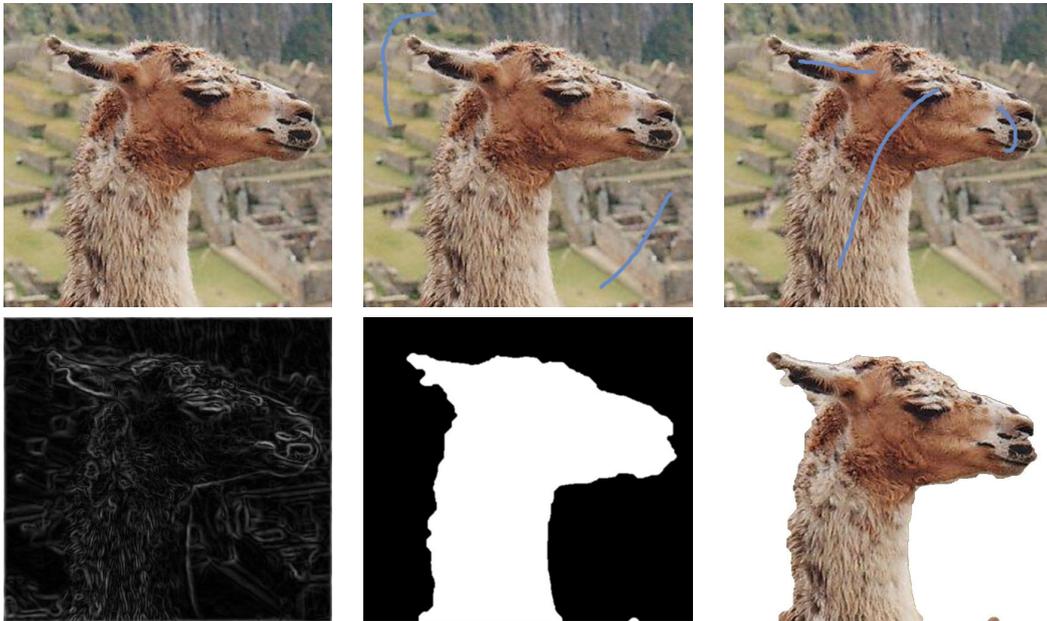


Figura 14: Segmentação de um objeto utilizando $ng_o = 10$ e $ng_f = 6$. A imagem possui 548x510 pixels e 24 bits.

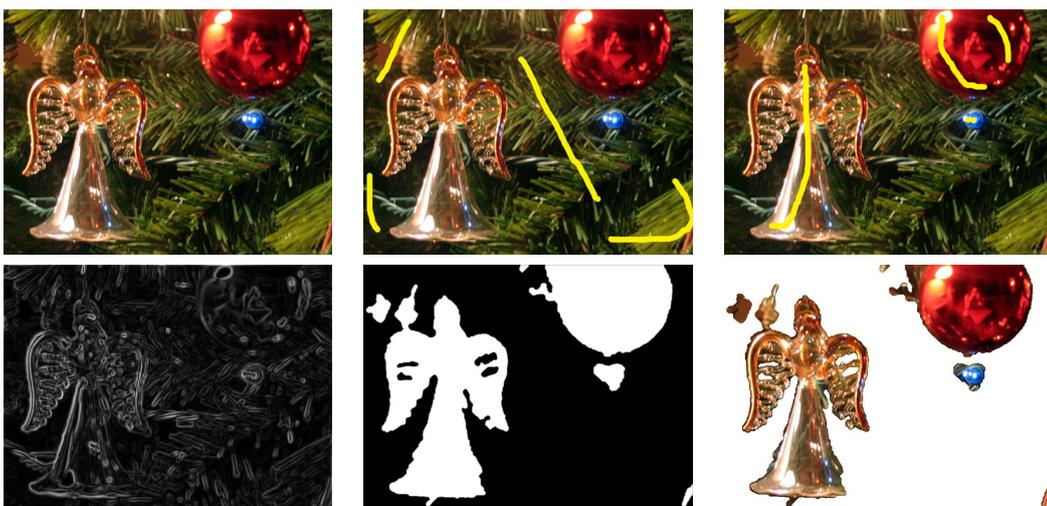


Figura 15: Segmentação de três objetos utilizando $ng_o = 6$ e $ng_f = 2$. A imagem possui 800x600 pixels e 24 bits.

A Tabela 1 contém as informações sobre o tempo gasto, em segundos, para construir o histograma do fundo e do objeto com as cores marcadas, realizar as tarefas de calcular o gradiente de Di Zenzo com o operador de Shen-Castan, as GMM, minimizar a energia, e a porcentagem desses tempos em relação ao tempo total.

Tabela 1: *Tempo de Execução de cada etapa do algoritmo*

Figura	Tamanho	Histograma	Shen-Castan e Di Zenzo	GMM	Minimização	Total
10	800x600	2s(2,53%)	8s(10,13%)	62s(78,48%)	3s(3,79%)	79s
11	636x876	0s(0%)	9s(16,36%)	38s(69,09%)	3s(5,45%)	55s
12	800x549	1s(2,38%)	7s(16,66%)	27s(64,28%)	2s(4,76%)	42s
13	548x510	0s(0%)	4s(14,28%)	19s(67,86%)	1s(3,57%)	28s
14	548x510	0s(0%)	4s(5,97%)	58s(86,56%)	2s(2,98%)	67s
15	800x600	2s(3,03%)	8s(12,12%)	48s(72,72%)	3s(4,54%)	66s

Com a Tabela 1, pode-se concluir que a etapa mais custosa do método apresentado neste trabalho é o cálculo das misturas gaussianas. Nesta etapa, o tamanho da imagem e o número de gaussianas influenciam muito o tempo de execução. Por exemplo, a Figura 10, por ser maior, gastou mais tempo para o cálculo da GMM do que a Figura 13. Já a Figura 14, por ter usado mais gaussianas, levou um tempo muito maior do que a Figura 13, mesmo ambas sendo iguais.

O cálculo do gradiente de Di Zenzo só sofreu influência do tamanho da imagem. A minimização e o cálculo do histograma foram, em geral, as etapas mais rápidas.

Conclui-se que a energia criada neste trabalho é muito dependente das misturas gaussianas, o que deixou o método lento.

Além do tempo de execução, outro problema deste método é quando as cores do fundo e do objeto coincidem ou são muito próximas. Isso levou a alguns erros de rotulação, como pode ser visto na Figura 15, onde parte do fundo foi confundida com objeto. Isso também levou a problemas nas bordas dos objetos, como na Figura 11, o que pode prejudicar o uso da segmentação em outras tarefas, por exemplo, a reconstrução de documentos fragmentados faz uso das bordas do objeto para a junção das peças.

Apesar destes problemas, o método se mostrou eficiente não só para imagens com pouca variação de cor no fundo, como as Figuras 10 e 11, mas também imagens mais complexas, como as Figuras 14 e 15.

4 *Conclusão*

Neste trabalho foram apresentados os conceitos necessários para segmentação de imagens como um problema de otimização. Diferentes trabalhos nesta área foram estudados, como o caso da otimização por corte em grafos e das técnicas utilizando multiresolução. Optou-se pelo uso de corte em grafos para a minimização da energia para o encontro da função característica da imagem.

Uma atenção maior foi dada à criação de uma energia que levasse em consideração as informações da GMM e das altas frequências da imagem. Isso foi possível utilizando a Energia de Gibbs. As informações da GMM foram usadas para calcular a penalidade do pixel ser do fundo ou do objeto, uma vez que ela mostra uma boa distribuição de probabilidade das cores. As informações de altas frequências, provenientes do gradiente de Di Zenzo, ajudaram a aumentar a energia das bordas no cálculo do custo da penalidade para que dois elementos vizinhos pertençam a conjuntos diferentes.

É importante destacar que o algoritmo ficou robusto o suficiente para segmentar imagens mais complexas. Portanto, também o é para a segmentação de imagens com fundo com pouca variação de cor, objetivo principal desta monografia.

O grande problema do método apresentado é sua alta complexidade de tempo e espaço, a qual pode ser verificada na Tabela 1. Portanto, técnicas mais eficientes devem ser implementadas.

As noções de multiresolução não puderam ser aplicadas em tempo hábil, mas algumas idéias já foram estudadas para um futuro artigo e/ou dissertação de mestrado. Uma delas seria que o peso α das componentes gaussianas de cada GMM leve em consideração sua importância em cada resolução da imagem. Desta forma, as gaussianas que têm distribuição de cores que aparecem em mais de uma resolução, tem maior peso para o cálculo das penalidades.

Como trabalhos futuros, pode-se utilizar multiresolução e o segmentador para auxílio de outras tarefas, por exemplo, a reconstrução automática de quebra-cabeças, onde a

segmentação faria a extração das bordas para posterior junção das peças.

Referências

- BOCCIGNONE, G. et al. A multiresolution diffused expectation-maximization algorithm for medical image segmentation. *Comput. Biol. Med.*, Pergamon Press, Inc., Elmsford, NY, USA, v. 37, n. 1, p. 83–96, 2007. ISSN 0010-4825.
- CARSON, C. et al. Blobworld: Image segmentation using expectation-maximization and its application to image querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 24, p. 1026–1038, 2002.
- CHOI, H.; BARANIUK, R. G. Multiscale image segmentation using wavelet-domain hidden markov models. *IEEE Trans. on Image Processing*, v. 10, p. 1309–1321, 2001.
- KOEPFLER, G.; LOPEZ, C.; MOREL, J. M. A multiscale algorithm for image segmentation by variational method. *SIAM J. Numer. Anal.*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, v. 31, n. 1, p. 282–299, 1994. ISSN 0036-1429.
- KOLMOGOROV, V.; ZABIN, R. What energy functions can be minimized via graph cuts? *Pattern Analysis and Machine Intelligence*, v. 26, p. 147–159, 2004.
- LI, Y. et al. Lazy snapping. Siggraph, 2004.
- RAJA, Y.; MCKENNA, S. J.; GONG, S. Segmentation and tracking using colour mixture models. In: *In Asian Conference on Computer Vision*. [S.l.: s.n.], 1998. p. 607–614.
- REIGOTA, N. *Comparação da transformada de wavelet discreta e da transformada do cosseno para compressão de imagens de impressão digital*. 161 p. Dissertação (Mestrado) — USP, 2007.
- RODRIGUES, D. B. *Segmentação de Objetos Matriciais por Corte em Grafos*. 2007. Universidade Federal de Juiz de Fora.
- ROTHER, C.; KOLMOGOROV, V.; BLAKE, A. Grabcut - interactive foreground extraction using iterated graph cuts. Siggraph, 2004.
- SHEN, J.; CASTAN, S. Edge detection based on multiedge models. *Real-Time Image Processing: Concepts and Technologies, SPIE Proceedings*, Society for Photo-Optical Instrumentation Engineers, Bellingham, WA, v. 860, 1987.
- ZENZO, S. D. A note on the gradient of a multi-image. *Comput. Vision Graph. Image Process.*, Academic Press Professional, Inc., San Diego, CA, USA, v. 33, n. 1, p. 116–125, 1986. ISSN 0734-189X.