

3

Segmentador Multiresolução Estendido

Neste capítulo é abordado o método de segmentação de imagens proposto neste trabalho. A seção 3.1 apresenta o método de segmentação. Na seção 3.2 são apresentados os atributos de forma utilizados no método. E por fim, a seção 3.3 apresenta a protótipo desenvolvido.

3.1.

Processo de Segmentação

O método de segmentação de imagens proposto neste trabalho é uma extensão do algoritmo de crescimento de regiões apresentado por Baatz e Schäpe (Baatz00) e apresentado com detalhes na seção 2.1.4.

No início do processo de segmentação cada pixel da imagem é inicializado como uma semente, representando um objeto. Após a inicialização, um processo iterativo tem início. Em cada iteração, cada objeto é visitado uma única vez de uma forma pseudo-aleatória, garantindo assim o crescimento distribuído dos objetos.

Ao ser visitado, cada objeto identifica qual dos objetos vizinhos representa o menor acréscimo de heterogeneidade. Se a heurística de decisão escolhida for Melhor Vizinho (BestFitting), a fusão é realizada. Se a heurística escolhida for Melhor Vizinho Mútuo (MutualFitting), é verificado se essa condição é mútua, ou seja, se o objeto visitado também representa o menor acréscimo de heterogeneidade para o objeto vizinho. Nesta heurística, a fusão ocorre apenas se essas duas condições forem satisfeitas. Lembrando que segundo o algoritmo original, independente da heurística utilizada, uma fusão apenas ocorre se o fator da fusão (valor do acréscimo da heterogeneidade) for inferior ao quadrado do parâmetro de escala p_e^2 , que atua como um limiar global.

O critério de heterogeneidade é basicamente o mesmo apresentado na seção 2.1.4, sendo a formulação geral dada pela Equação 2-1. Esse critério possui uma componente espectral h_{cor} e uma morfológica h_{forma} . A componente espectral é

definida pelos valores dos pixels que compõem o objeto, sendo proporcional ao desvio-padrão desses valores, ponderados por pesos arbitrários definidos para cada banda espectral da imagem. A formulação da componente espectral é dada pela Equação 2-2. No algoritmo proposto optou-se por normalizar os pesos das bandas w_c , de forma que: $\sum_c w_c = 1$.

A componente morfológica é definida pelo desvio relativo da forma do objeto em relação a formas geométricas bem definidas. O algoritmo original considera dois atributos de forma: compacidade e suavidade. O método proposto neste trabalho estende o número de atributos de forma disponíveis para o cálculo da componente morfológica. A seção seguinte apresentará com detalhes os dois atributos usados no algoritmo original e os novos atributos sugeridos na extensão deste.

A formulação da componente morfológica h_{forma} é então generalizada onde, não apenas dois, mas vários atributos de forma podem ser utilizados. Na Equação 3-1, s e w_s são, respectivamente, o índice e o peso do atributo de forma; a_s é o valor do atributo de forma e n o número de pixels de cada objeto.

$$h_{forma} = \sum_s w_s (n_{obj3} \cdot a_s^{obj3} - (n_{obj1} \cdot a_s^{obj1} + n_{obj2} \cdot a_s^{obj2})) \quad 3-1$$

Os pesos dos atributos de forma w_s também são normalizados, de forma que: $\sum_s w_s = 1$.

A segmentação termina quando mais nenhuma fusão puder ser realizada. Os objetos finais representam o resultado da segmentação. Vale ressaltar que por conta da visitação dos objetos ser pseudo-aleatória, é garantida a reprodutibilidade dos resultados.

3.2. Atributos de Forma

O algoritmo original sugere dois atributos de forma para a composição da componente de heterogeneidade morfológica:

- a) **compacidade** – representa o desvio em relação a um quadrado (Baat00). É dado pela razão entre o comprimento de borda do objeto l e a raiz da sua área n . $a_{compacidade}$ é mínimo para um quadrado. É sensível a irregularidades de contorno.

$$a_{compacidade} = \frac{l}{\sqrt{n}} \quad 3-2$$

- b) **suavidade** – representa o desvio em relação a uma forma convexa. É dado pela razão entre o comprimento de borda do objeto l e o comprimento de borda do retângulo envolvente mínimo l_{rem} . $a_{suavidade}$ é mínimo para um retângulo perfeito e aumenta com a presença de concavidades. É sensível à suavidade do contorno. Adaptado de (Baat00), pois no algoritmo original, o retângulo envolvente é utilizado.

$$a_{suavidade} = \frac{l}{l_{rem}} \quad 3-3$$

Neste trabalho, propomos a extensão da componente de heterogeneidade morfológica pela adição de mais 8 atributos de forma. São eles:

- c) **retangularidade** – representa o desvio em relação a uma forma retangular (Halcon11, adaptado). É dado pela razão entre a área do retângulo envolvente mínimo n_{rem} e a área do objeto n . $a_{retangularidade}$ é igual a 1 para um retângulo perfeito e aumenta com a presença de reentrâncias. É sensível à presença de ramificações finas e longas.

$$a_{retangularidade} = \frac{n_{rem}}{n} \quad 3-4$$

- d) **isometria** – representa o desvio em relação a uma forma isométrica (Halcon11). É dado pela razão entre o semieixo principal a e

secundário b da elipse ajustada. $a_{isometria}$ é igual a 1 para um objeto isométrico e aumenta à medida que ele se alonga, sendo proporcional ao alongamento do objeto.

$$a_{isometria} = \frac{a}{b} \quad 3-5$$

- e) **anisometria** – representa o desvio em relação a uma forma não-isométrica. É dado pela razão entre o semieixo secundário b e principal a da elipse ajustada. $a_{anisometria}$ é igual a 1 para um objeto isométrico e diminui à medida que ele se alonga, sendo inversamente proporcional ao alongamento do objeto.

$$a_{anisometria} = \frac{b}{a} \quad 3-6$$

- f) **bulkiness** – representa o desvio em relação a um círculo com raio de dimensão intermediária entre o semieixo principal a e o semieixo secundário b da elipse ajustada (Halcon11).

$$a_{bulkiness} = \frac{\pi \cdot a \cdot b}{n} \quad 3-7$$

- g) **fator de estrutura** – representa o desvio em relação a um círculo com raio igual ao semieixo principal a da elipse ajustada (Halcon11). É mínimo para um círculo e aumenta à medida que o semieixo principal aumenta em relação à área.

$$a_{fator_estrutura} = \frac{\pi \cdot a^2}{n} - 1 \quad 3-8$$

- h) **excentricidade** – representa o desvio em relação a um círculo. $a_{excentricidade}$ é igual a 0 para um círculo perfeito e tende a 1 quando o alongamento tende ao infinito.

$$a_{excentricidade} = \sqrt{1 - \left(\frac{b}{a}\right)^2} \quad 3-9$$

- i) **roundness** – representa o desvio em relação a um círculo com diâmetro igual ao eixo principal $2a$ da elipse ajustada (Russ98, adaptado). $a_{roundness}$ é igual a 1 para um círculo perfeito e aumenta à medida que o eixo principal aumenta em relação à área. Desta forma, é mais sensível ao alongamento do objeto (Gomes05).

$$a_{roundness} = \frac{\pi \cdot (2a)^2}{4 \cdot n} \quad 3-10$$

- j) **fator de forma circular** – representa o desvio em relação a um círculo (Russ98). $a_{fator_circular}$ é mínimo para um círculo perfeito e aumenta à medida que o perímetro aumenta em relação à área. Entretanto, se o contorno do objeto é irregular, o perímetro pode aumentar ainda que a forma geral do objeto seja próxima de um círculo. Dessa maneira, o fator de forma circular é mais sensível a irregularidades de contorno (Gomes05).

$$a_{fator_circular} = \frac{p^2}{4 \cdot \pi \cdot n} \quad 3-11$$

3.3. Implementação

Aqui são descritas as principais características e parâmetros do protótipo em software desenvolvido neste trabalho.

3.3.1. Protótipo

Para avaliar o método proposto nessa dissertação (Seção 3.1), um protótipo, chamado XMRS (eXtended MultiResolution Segmentation), foi desenvolvido. O XMRS foi codificado na linguagem de programação C++ (Stroustrup97),

utilizando a biblioteca de classes Qt4 (Nokia11) e, para o suporte a imagens, a biblioteca TerraLib (Câmara08).

O programa recebe como entrada a imagem a ser segmentada e os parâmetros do algoritmo de segmentação. Opcionalmente, uma imagem com segmentos de referência (para a avaliação da segmentação) pode ser passada para o programa. A saída é uma imagem rotulada, onde se identificam os segmentos resultantes.

O protótipo não possui interface gráfica, devendo ser executado através de uma linha de comando, conforme a figura abaixo:

```
xmrs <img_entrada> <operação> <heurística> <pesos_bandas> <pesos_atrib_forma>  
<peso_cor_forma> <escala> <img_saída> <img_referência>
```

Figura 3.1: Linha de comando para a chamada do programa XMRS.

Os argumentos da linha de comando são os seguintes:

- a) **img_entrada** – caminho da imagem a ser segmentada. Suporta os formatos TIF e JPG.
- b) **operação** – modo de operação do segmentador. Aceita os seguintes valores:
 - a. **VisMean** – gera a imagem de saída com os objetos resultantes coloridos pelo valor médio das bandas.
 - b. **VisRandom** – gera a imagem de saída com os objetos resultantes coloridos aleatoriamente.
 - c. **VisImage** – gera a imagem de saída com o contorno dos objetos resultantes sobreposto à imagem de entrada.
 - d. **Eval** – não gera imagem de saída.
- c) **heurística** – heurística de decisão da segmentação. Aceita os seguintes valores:
 - a. **BestFitting** – o objeto é fundido com o vizinho que represente o menor acréscimo de heterogeneidade.
 - b. **MutualFitting** – o objeto é fundido com o seu melhor vizinho apenas se a relação for mútua.

- d) **pesos_bandas** – pesos das bandas. Os pesos são normalizados de forma que a sua soma seja igual a 1,0.
- e) **pesos_atrib_forma** – pesos dos atributos de forma. Os pesos são normalizados de forma que a sua soma seja igual a 1,0.
- f) **peso_forma_cor** – importância relativa entre forma e cor. Aceita valores no intervalo [0,1]. O valor 0,0 indica que apenas a cor será considerada. O valor 1,0 indica que apenas a forma será considerada, mas é aconselhável que não ultrapasse 0,9, já que a forma sem a informação espectral é pouco útil (Baatz00).
- g) **escala** – parâmetro de escala. Aceita valores no intervalo [0,∞].
- h) **img_saída** – nome da imagem de saída, gerada em formato TIF.
- i) **img_referência (opcional)** – imagem com segmentos de referência. Deve ser uma imagem RGB em formato TIF onde cada segmento de referência possui uma cor distinta. Caso seja definida, além da segmentação o XMRS também irá avaliar a qualidade da segmentação utilizando a métrica RBSB definida na seção 2.4.1.

3.3.2. Ambiente

O XMRS foi programado e testado em ambiente Windows, mas o seu código-fonte é multiplataforma. Portanto, nada impede que seja compilado e utilizado em outros sistemas operacionais como o Linux.