

Avaliação de superpixels para segmentação e detecção de contorno de imagens

Felipe Augusto Lima Reis¹

¹PUC Minas - Pontifícia Universidade Católica de Minas Gerais

`falreis@sga.pucminas.br`

Abstract. *Superpixels are structures that group similar pixels into sets that reflect aspects of the image. This paper evaluates the use of SLIC and EGB superpixels for segmentation. It's also evaluates the combination of both methods. Finally, this paper shows an hierarchical method implementation, using SLINK (single-linkage) clusterization, applied for the superpixels methods. For evaluation, this paper uses the Berkeley Segmentation Data Set (BSDS500). The results were compared to the dataset ground truth, using precision and recall.*

Resumo. *Superpixels são estruturas que agrupam pixels semelhantes em conjuntos que refletem aspectos da imagem. Este artigo avalia a utilização de superpixels SLIC e EGB para segmentação. Avalia também os benefícios da combinação dos métodos para produção de segmentação. Por fim, o artigo mostra uma implementação de método hierárquico, utilizando clusterização SLINK (single-linkage), para os métodos estudados nesse trabalho. Para avaliação dos resultados, foram utilizadas as imagens do conjunto de validação do Berkeley Segmentation Data Set (BSDS500). Os resultados foram comparadas em relação ao ground truth, utilizando o método de precisão e revocação.*

1. Introdução

A segmentação consiste em dividir uma imagem em um conjunto de regiões logicamente agrupadas, de modo a reunir áreas que contém informação relevante dentro dos grupos, idealmente correspondente a objetos reais [Domínguez and Morales 2016] [Zhang et al. 2008]. A segmentação é utilizada no processamento de imagens, vídeos e aplicações de visão computacional. Também consiste em um importante passo na tentativa de explicar uma imagem por meio de algoritmos [Zhang et al. 2008]. Extensiva pesquisa é realizada e muitas abordagens e algoritmos são utilizados, com bons resultados para um conjunto ou classes de imagens [Zhang et al. 2008].

Para tarefas de segmentação, podemos tomar os *pixels* como unidades básicas de processamento [Wang et al. 2017a]. O agrupamento de pixels em unidades maiores permite um tipo de segmentação chamado de *oversegmentation* [Wang et al. 2017a], ilustrado na Figura 1. Alguns métodos de geração de superpixels são utilizados para segmentação de imagens e detecção de bordas, como os métodos EGB [Felzenszwalb and Huttenlocher 2004] e SLIC [Achanta et al. 2012]. Esse trabalho investiga se a utilização de métodos segmentação e detecção de contornos baseados em superpixels. Também investiga o agrupamento de superpixels e a criação de hierarquias, permitindo maior ou menor nível de detalhamento para uma representação.

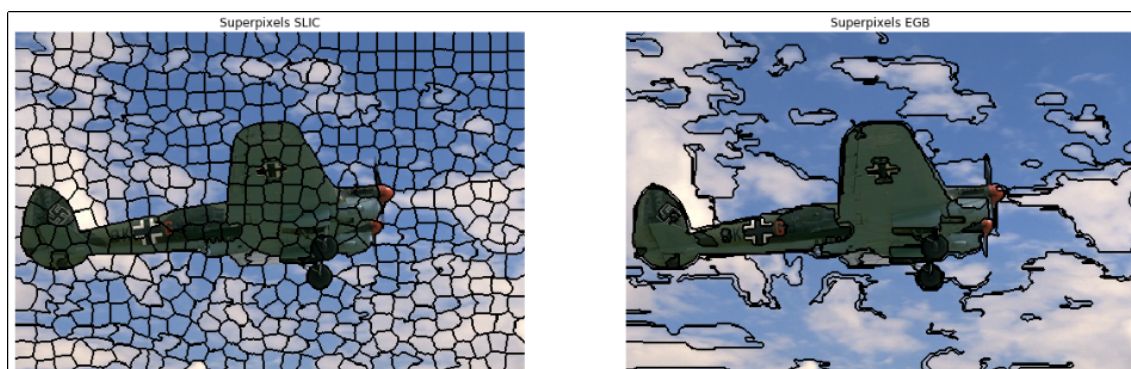


Figura 1. Imagens segmentadas utilizando superpixels SLIC e EGB

O presente trabalho apresenta a seguinte estrutura: a Seção 2 mostra o referencial teórico para construção do trabalho, a Seção 3, exhibe os métodos desenvolvidos, algoritmos e os critérios para elaboração dos testes; a Seção 4 mostra os resultados obtidos nos testes realizados e a discussões dos mesmos; a Seção 5 contém a conclusão do artigo, com as considerações finais.

2. Referencial Teórico

2.1. Superpixels

Superpixels são estruturas que agrupam pixels semelhantes em conjuntos. O agrupamento possibilita a redução de complexidade das tarefas de processamento [Achanta et al. 2012], ao reduzir a quantidade de itens a serem processados. Os superpixels são utilizados na área de visão computacional para solução de vasto número de problemas, como detecção de contorno [Wang et al. 2017b], segmentação [Liu et al. 2016] e localização de objetos [Fulkerson et al. 2009].

Superpixels, segundo [Felzenszwalb and Huttenlocher 2004], devem capturar importante grupos ou regiões, refletindo aspectos da imagem. Devem também ser executados em tempo próximo ao linear em relação a quantidade de pixels. Existem diversas abordagens para a geração de superpixels [Achanta et al. 2012]. Podemos classificá-las, segundo o método de agrupamento, em dois grandes grupos [Achanta et al. 2012]:

- *Algoritmos baseados em grafos*: tratam cada pixel como um nó do grafo. Os pesos das arestas são proporcionais a similaridade entre os pixels vizinhos [Achanta et al. 2012]. Dentre os algoritmos baseados em grafos podemos citar o *Efficient Graph-Based Image Segmentation* (EGB) [Felzenszwalb and Huttenlocher 2004];
- *Algoritmos baseados em gradiente ascendente*: partem de um conjunto inicial de pixels, e utilizam métodos de gradiente ascendente iterativamente para refinar os conjuntos (*clusters*) até que se atinja um critério de convergência [Achanta et al. 2012]. Nesse conjunto, podemos citar as abordagens *watersheds* [Vincent and Soille 1991] e o algoritmo SLIC (*Simple Linear Iterative Clustering*) [Achanta et al. 2012]

2.1.1. Superpixels SLIC e SLICO

O algoritmo SLIC utiliza um único parâmetro k , correspondente a quantidade aproximada de superpixels, gerados em formato regular [Achanta et al. 2012]. A Figura 2 ilustra o algoritmo SLIC para diferentes números de superpixels. A fim de produzir tamanhos semelhantes, o intervalo de busca analisado é $S = \sqrt{N/k}$, onde N é o número de pixels da imagem [Achanta et al. 2012].

A definição do centro dos superpixels é feita utilizando sementes, que são movidas para locais de geração. Esses locais correspondem a posição mais baixa do gradiente em uma vizinhança de 3×3 [Achanta et al. 2012]. Esse passo evita que superpixels sejam centrados nas bordas ou em uma posição de ruído [Achanta et al. 2012]. Cada pixel, então, é associado com o centro do cluster mais próximo, de modo que as regiões de busca se sobreponham [Achanta et al. 2012]. A fim de aumentar o desempenho do algoritmo, a região de busca é limitada em 2 vezes o tamanho aproximado do superpixel S , gerando busca em uma área $2S \times 2S$ [Achanta et al. 2012]. Em seguida, um passo atualiza os centros dos clusters e computa o erro residual E [Achanta et al. 2012]. O Algoritmo 1, resume as informações descrita nesse parágrafo.

Algorithm 1 Segmentação por superpixels SLIC (*Adaptado de* [Achanta et al. 2012])

```
1: /* Inicialização */
2: Inicialize os centros dos clusters  $C_k = [l_k, a_k, b_k, x_k, y_k]^T$  por amostragem de pixels
   em etapas de grade regulares  $S$ 
3: Mova os centros dos clusters para a posição de menor gradiente em uma vizinhança
    $3 \times 3$ 
4: Faça label  $l(i) = -1$  para cada pixel  $i$ 
5: Faça distância  $d(i) = \infty$  para cada pixel  $i$ 
6: repeat
7:   for cada cluster centrado em  $C_k$  do
8:     for cada pixel  $i$  em uma região  $2S \times 2S$  ao redor de  $C_k$  do
9:       Compute a distância  $D$  entre  $C_k$  e  $i$ 
10:      if  $D < d(i)$  then
11:        faça  $d(i) = D$ 
12:        faça  $l(i) = k$ 
13:      end if
14:    end for
15:  end for
16:  /* Atualização */
17:  Compute novos centros dos clusters
18:  Compute o erro residual  $E$ 
19: until  $E \leq threshold$ 
```

Para compreensão mais precisa do Algoritmo 1, é necessário entender o método para cálculo da medida de distância D entre os conjuntos. O algoritmo SLIC trabalha no espaço de cores *CIE LAB* junto ao espaço-plano *labxy*. Nesse cenário, a posição do pixel pode assumir um intervalo de valores, fazendo com que o cálculo da distância não possa ser feito utilizando uma distância euclidiana; é necessária uma prévia normalização

da proximidade espacial e de cor.

No espaço de cores CIELAB, cada pixel de cores é representado por $[lab]^T$, sendo que uma grande variedade de valores é possível [Achanta et al. 2012]. A posição do pixel $[xy]^T$, por sua vez, pode assumir um conjunto de valores de acordo com o tamanho da imagem [Achanta et al. 2012]. A definição simples de D como uma distância Euclidiana de 5 dimensões no espaço $labxy$, pode causar inconsistências no comportamento dos clusters para diferentes tamanhos de superpixels [Achanta et al. 2012]. Em superpixels grandes, as distâncias espaciais superam a proximidade da cor, dando mais importância relativa à proximidade espacial do que à cor [Achanta et al. 2012], fazendo com que os superpixels não sejam aderentes as bordas. Para solucionar o problema, é necessário normalizar a proximidade espacial e de cores, com as respectivas distâncias máximas dentro do cluster N_s e N_c . Com isso, considerando $d_c = \sqrt{(l_j - l_i)^2 + (a_j - a_i)^2 + (b_j - b_i)^2}$ a distância entre as cores e $d_s = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$, a distância espacial, temos:

$$D' = \sqrt{\left(\frac{d_c}{N_c}\right)^2 + \left(\frac{d_s}{N_s}\right)^2} \quad (1)$$

Como a distância espacial máxima esperada dentro de um cluster corresponde ao tamanho da amostra $N_s = S = \sqrt{N/K}$. Considerando também que a distância máxima de cores N_c , pode variar de cluster para cluster, podemos fixar N_c , como uma constante m e reescrever a Equação 1 [Achanta et al. 2012]:

$$D' = \sqrt{\left(\frac{d_c}{m}\right)^2 + \left(\frac{d_s}{S}\right)^2} \quad (2)$$

A Equação 2, por sua vez, pode ser simplificada na forma adotada pelos autores, como na Equação 3:

$$D = \sqrt{d_c^2 + \left(\frac{d_s}{S}\right)^2 \cdot m^2} \quad (3)$$

Uma etapa extra no processo de pós processamento é a união de *pixels orfãos*. Esses pixels são adicionados ao cluster mais próximo usando o algoritmo de componentes conexos [Achanta et al. 2012].

A complexidade do algoritmo SLIC é $O(n)$, devido a limitação do espaço de pesquisa do algoritmo SLIC [Achanta et al. 2010]. Outros algoritmos que utilizam *k-means* para segmentação tem custo $O(kNI)$, onde N é o número de pontos (pixels na imagem), k corresponde ao número de clusters (ou sementes) e I corresponde ao número de iterações para convergência [Achanta et al. 2010]. A modificação no algoritmo SLIC está na redução de cálculos redundantes de distância [Achanta et al. 2010].

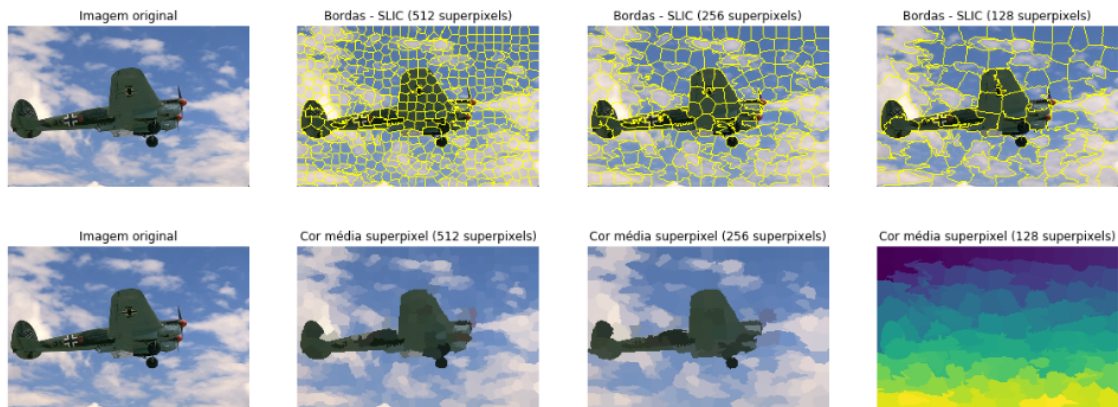


Figura 2. Fronteiras e coloração pelo valor médio dos superpixels SLIC/SLICO, para diferentes quantidades de superpixels

2.1.2. Superpixels EGB

Os superpixels EGB (*Efficient Graph-Based Image Segmentation*) utilizam uma abordagem baseada em grafos não direcionados. Nessa abordagem, cada *pixel* corresponde a um nó do grafo e a ligação entre eles ocorre por meio de arestas, com pesos não negativos, correspondente a medida de dissimilaridade [Felzenszwalb and Huttenlocher 2004].

Na abordagem baseada em grafos, a segmentação S é uma partição dos vértices V em componentes, no qual cada região $C \in S$ corresponde a um componente conectado em um grafo $G' = (V, E')$, onde $E' \subseteq E$, ou seja, a segmentação é induzida por um conjunto de vértices em arestas E [Felzenszwalb and Huttenlocher 2004].

No algoritmo foi definido um predicado D para avaliação da evidência de bordas entre dois componentes de uma segmentação. O algoritmo avalia a dissimilaridade entre elementos de dois componentes e os compara com elementos vizinhos em um mesmo componente, de modo que o algoritmo possa se adaptar em relação às características dos dados [Felzenszwalb and Huttenlocher 2004].

Para a comparação entre as regiões é utilizada uma função de corte (*threshold*) τ , a fim de medir o grau de diferença entre os componentes. Esse grau deve ser superior à diferença interna mínima, evidenciando uma borda [Felzenszwalb and Huttenlocher 2004]. A função de corte, no algoritmo, é utilizada baseada no tamanho do componente $\tau = k/|C|$, onde $|C|$ corresponde ao tamanho do componente C e k corresponde a um parâmetro do algoritmo [Felzenszwalb and Huttenlocher 2004].

O algoritmo EGB, utilizando pesos inteiros e ordenação por contagem, pode ser executado com custo linear, com complexidade $O(n \log n)$, para qualquer método de ordenação [Felzenszwalb and Huttenlocher 2004].

2.2. Análise de Clusters

Análise de cluster é a tarefa de agrupar dados em grupos (*clusters*) que tenham características em comum, sejam significativos e/ou úteis [Jeyalakshmi and Shanmugapriya 2016]. *Clusters* hierárquicos correspondem a análise

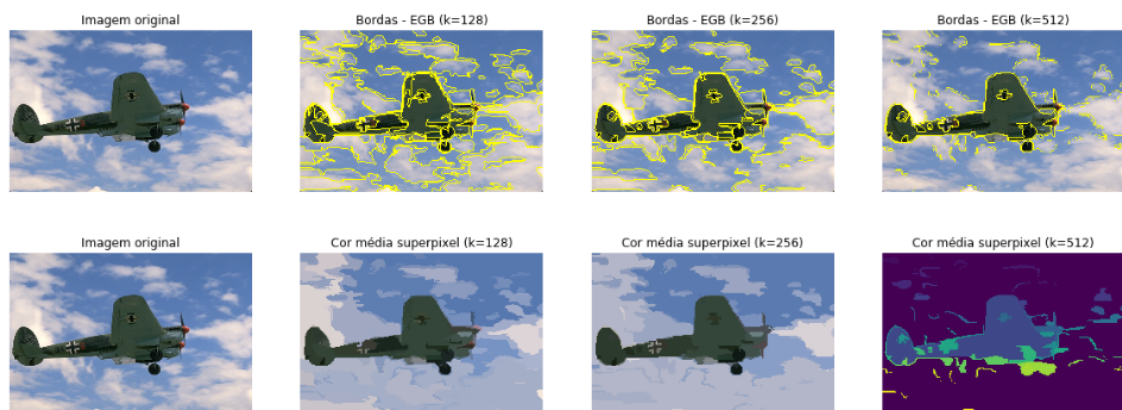


Figura 3. Fronteiras e coloração pelo valor médio dos superpixels EGB, para diferentes quantidades de superpixels

de *clusters* a fim de buscar hierarquias entre eles. As estratégias para hierarquização de cluster se dividem em dois grupos [Rokach and Maimon 2005]:

- Aglomerativo - abordagem "*bottom up*", em que a observação inicia-se no próprio *cluster* e os pares de *clusters* são unidos na medida em que se sobe na hierarquia.
- Divisivo - abordagem "*top down*" onde as observações iniciam-se em um *cluster* e são realizadas recursivamente a medida em que se desce na hierarquia.

Os algoritmos de clusterização hierárquicos aglomerativos, podem ser classificados, em geral, como gulosos [Murtagh and Contreras 2012]. Uma sequência de passos irreversíveis são usados para construção dos dados desejados [Murtagh and Contreras 2012]. A abordagem de clustering hierárquico de ligação única (*single-linkage*) produz um conjunto de clusters em cada nível - ou para cada valor de limite que produz uma nova partição [Murtagh and Contreras 2012]. O nome ligação única surge quando a dissimilaridade de interconexão entre dois grupos ou componentes é definida como a menor diferença de interconexão entre um membro de um e um membro do outro [Murtagh and Contreras 2012].

Os algoritmos originais de clusterização possuem complexidade $O(n^3)$. Alguns algoritmos de *single-linkage*, entretanto, como o SLINK, possuem complexidade $O(n^2)$ [Sibson 1973] [Murtagh and Contreras 2012]. O SLINK utiliza a distância mínima como critério de ligação (*linkage*) entre os *clusters*. A fórmula de dissimilaridade de Lance-Williams $\min\{d_{ik}, d_{jk}\}$, corresponde a distância mínima para dois pares de clusters i e j observados, onde d é a métrica de distância escolhida [Murtagh and Contreras 2012].

A partir da aglomeração de clusters é possível construir uma árvore hierárquica binária, onde os clusters são agrupados por sua similaridade. Esse tipo de árvore binária para visualização também é conhecido como dendrograma [Murtagh and Contreras 2012].

2.3. Precisão e Revocação

Em um problema de decisão, os classificadores podem ser positivos ou negativos [Davis and Goadrich 2006]. A decisão feita por um classificador pode ser representada por uma estrutura conhecida como matriz de confusão ou tabela de contingência. Essa matriz pode identificar 4 tipos de dados possíveis [Davis and Goadrich 2006]:

- Verdadeiros Positivos - correspondem aos valores que foram corretamente classificados como positivos;
- Falsos Negativos - correspondem aos valores que foram classificados incorretamente como negativos;
- Falsos Positivos - correspondem aos valores que foram classificados incorretamente como positivos;
- Verdadeiros Negativos - correspondem aos valores que foram classificados corretamente como negativos [Davis and Goadrich 2006];

O método de precisão e revocação é um método de classificação binária, onde a precisão (*prediction*) mede a proporção de eventos do modelo que são reais; enquanto a revocação (*recall*), mede a proporção de eventos ocorridos no domínio que são capturados pelos modelos [Torgo and Ribeiro 2009].

A partir dos valores de precisão e revocação, é possível construir curvas que associem ambos os valores. Essas curvas possibilitam uma informação visual da performance de um algoritmo [Davis and Goadrich 2006].

A classificação de precisão e revocação pode ser dada por uma média harmônica, chamada de *F-measure* [Dembczyński et al. 2011]. Essa medida é utilizada para diferentes problemas de classificação e impõe um melhor equilíbrio entre o desempenho entre precisão e revocação, respectivamente, e, portanto, é mais adequado no caso de dados desequilibrados [Dembczyński et al. 2011].

Dado um vetor binário m -dimensional e classes de precisão e revocação, cujo conjunto estejam associados ao vetor m -dimensional, a medida F-measure é dada pela Equação 4, correspondente a média harmônica entre ambas as medidas [Dembczyński et al. 2011].

$$F = \frac{2 \sum_{i=1}^m prec \cdot revoc}{\sum_{i=1}^m prec + \sum_{i=1}^m revoc} \in [0, 1] \quad (4)$$

2.4. Hierarquia de Segmentações

A hierarquização das segmentações tem como objetivo possibilitar a utilização dos algoritmos para produzir segmentação em diferentes níveis de detalhamento: desde um nível mais aprofundado até um nível macro. Os diferentes níveis hierárquicos podem ser combinados em termos do contorno das hierárquias, permitindo a representação das hierárquias indexadas como bordas suaves, chamadas Mapas de Contorno Ultramétrico (UCM - *Ultrametric Contour Map*) [Arbelaez 2006].

Para que os níveis hierárquicos possam garantir a manutenção da informação, agrupando apenas características semelhantes, as hierárquias devem seguir os princípios de análise multiescala [Guimarães and Patrocínio 2013]. Esses princípios asseguram manutenção de duas características principais [Guimarães and Patrocínio 2013]:

- Causalidade - um contorno presente em uma escala $k1$ deve estar presente em qualquer escala $k2 < k1$ [Guimarães and Patrocínio 2013];

- Localidade - a medida em que o número de regiões diminui, os contornos devem ser estáveis (não devem se mover). A união corresponde a manutenção das bordas dos grupos que se fundiram [Guimarães and Patrocínio 2013].

3. Métodos

Para confecção do trabalho foram escolhidos os algoritmos de SLIC e EGB. Os algoritmos possuem características diferentes: o SLIC é capaz de produzir *superpixels* em formas regulares, porém não é tão preciso ao separar os conjuntos por similaridade; por outro lado, o EGB, produz *superpixels* irregulares, porém é mais aderente às diferenças de cores entre *pixels*.

Nos testes realizados foi realizada hierarquização dos resultados obtidos após a geração de *superpixels* SLIC, EGB e também sobre a composição SLIC+EGB. A composição SLIC+EGB corresponde a aplicação do algoritmo SLIC, seguido pela recoloração utilizando o valor médio de cores do *superpixel* e, por fim, a aplicação do método EGB. A partir dessa seção, o método SLIC+EGB passará a ser chamado SGB, para facilitar a nomenclatura.

Os resultados das hierarquizações foram comparados aos melhores resultados de segmentação obtidos pela segmentação SLIC, EGB e SGB sem hierarquização. O método de hierarquização está descrito na seção 3.1, enquanto o método de avaliação de resultados está descrito na seção 3.3. Os resultados dos testes estão disponíveis na seção 4.

3.1. Hierarquias

Para construção das hierarquias foram gerados *clusters* com as correlações entre os grupos de *superpixels* adjacentes, após a segmentação inicial. Foi utilizado o algoritmo SLINK (*single-linkage*) para agrupamento dos *superpixels* próximos, gerando uma árvore com as correlações entre os *superpixels*.

Para verificação da estrutura da árvore foi utilizado um dendrograma, como representado na Figura 4. O dendrograma é um tipo de árvore binária utilizado para ilustração de uma clusterização hierárquica [Murtagh and Contreras 2012].

Em um árvore de estrutura semelhante à representada por um dendrograma é possível realizar cortes horizontais ou verticais na hierarquia, a fim de obter tipos de agrupamentos diferentes. No trabalho realizado somente foram realizados cortes horizontais, em diversos níveis.

Devido às características da segmentação, quando comparados ao *ground truth*, não foram validados todos os níveis hierárquicos. A fim de reduzir o número de comparações, somente foram avaliados níveis hierárquicos intermediários que apresentaram melhores resultados em testes empíricos. Com isso, foi estabelecido um limite máximo e mínimo para os cortes na hierarquia, de modo que os cortes foram feitos dentro de um intervalo, evitando avaliação de níveis com baixa probabilidade adequação ao *ground truth*.

O método de hierarquização descrito nesse trabalho manteve as características da análise multiescala. A Figura 5 mostra os possíveis cortes na hierarquia de uma segmentação SGB, sem recoloração da imagem. Nesse modelo, a segmentação é capaz de unir *pixels* por similaridade.

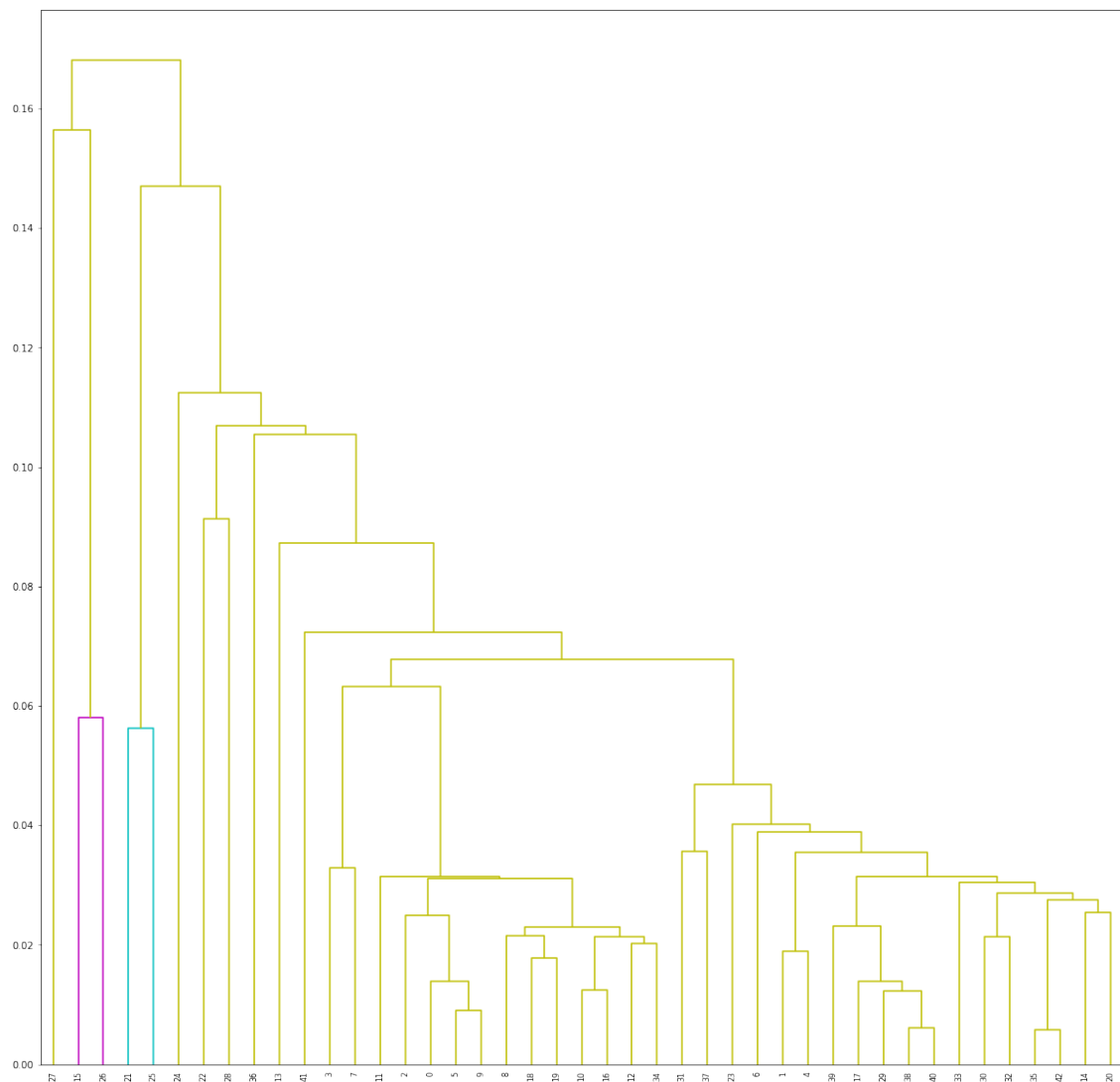


Figura 4. Dendrograma dos superpixels de uma imagem

É importante salientar que os métodos originais (SLIC e EGB), quando apenas variados os parâmetros de criação de superpixels não mantêm as características de análise multiescala, conforme podemos ver nas Figuras 2 e 3; essas características somente são obtidas pelos processos hierárquicos aplicados aos métodos após a criação dos superpixels.

3.2. Base de Dados

A fim de facilitar a pesquisa, alguns trabalhos foram desenvolvidos para criação de um conjunto de imagens com suas respectivas segmentações manuais. A base BSDS500 (*Berkeley Segmentation Data Set and Benchmarks 500*) provê uma base de imagens para pesquisa de segmentação e detecção de bordas. [Arbelaez et al. 2011]. Essa base de dados é uma extensão da base BSDS300, com 200 novas imagens para avaliação [Arbelaez et al. 2011].

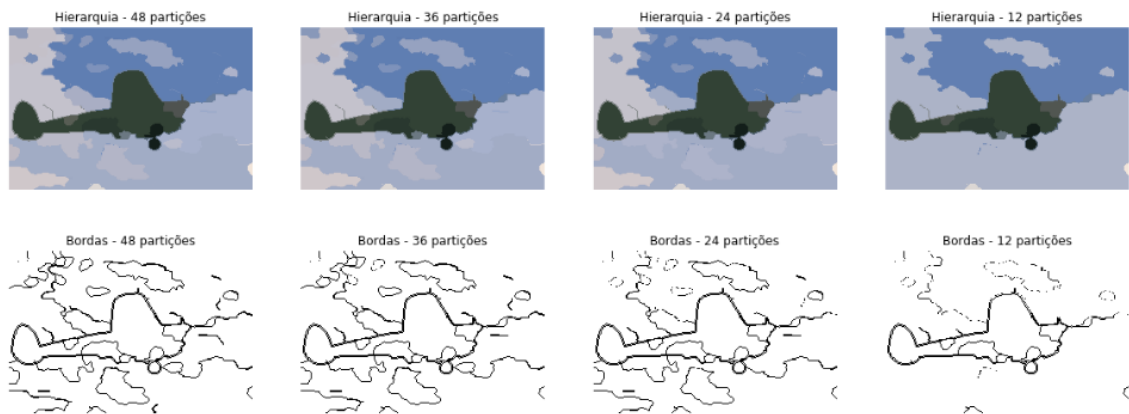


Figura 5. Hierarquia de partições utilizando superpixel SLIC+EGB.

3.3. Avaliação de Resultados

A fim de avaliar a efetividade das segmentações, tradicionalmente são utilizados métodos subjetivos, como a visualização humana ou métodos supervisionados, onde uma segmentação é comparada, de forma automatizada, a uma imagem manualmente segmentada [Zhang et al. 2008].

Para avaliação das imagens, foi primeiramente utilizada a classificação visual, comparando a qualidade dos resultados. Em seguida utilizou-se um método de comparação de precisão e revocação, para detecção de contornos e bordas, conforme detalhado na Seção 2.3.

A avaliação de contornos e bordas foi feito utilizando o benchmarks de avaliação de performance das segmentações, disponíveis na base de dados BSDS500¹.

3.4. Código Fonte e Bibliotecas Utilizados

Os códigos fontes gerados, realizando as segmentações, hierarquias, geração de mapas ultramétricos, avaliação dos resultados e os gráficos presentes nesse trabalhos estão disponíveis publicamente na página pessoal do autor no Github ².

Os algoritmos desenvolvidos para esse trabalho foram programados em Python, com auxílio da ferramenta web open-source Jupyter Notebook. As estrutura de dados utilizadas foram *arrays*, matrizes e listas, disponíveis na linguagem e adequadas ao objetivo do código fonte. As bibliotecas utilizadas foram gerenciadas pelo ambiente virtual Anaconda. O arquivo contendo as bibliotecas necessárias para configuração do ambiente está disponível na página pessoal do autor no Github.

Os algoritmos SLIC e EGB utilizados para realização desse trabalho foram obtidos pela biblioteca Scikit-Image³. Ambos os algoritmos foram desenvolvidos em Python, utilizando *arrays* em sua implementação. Para melhor desempenho, os códigos em Python foram compilados utilizando compilador C++ por meio do plugin Cython. A biblioteca

¹<https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html>

²<https://github.com/falreis/image-segm>

³<https://scikit-image.org/docs/dev/api/skimage.segmentation.html>

Parâmetros	SLIC	EGB	SGB	HSLIC	HEGB	HSGB
k	300	-	1408	300	-	1408
$scale$	-	300	1408	-	300	1408
σ	-	0.8	0.8	-	0.8	0.8
min_size	-	30	30	-	30	30
<i>partições avaliadas</i>	-	-	-	150	150	150

Tabela 1. Parametrização dos algoritmos.

utilizada já contém os códigos compilados, disponíveis na página oficial da biblioteca no Github ⁴.

Os algoritmos de geração de cluster de ligação única (*single-linkage*) e representação de dendrogramas foram obtidos na biblioteca Scipy.org⁵. A implementação, assim como os algoritmos SLIC e EGB, são desenvolvidos em Python e compilados usando Cython, para melhor desempenho. Utilizam arrays multidimensionais *numpy.ndarray* e também pilhas explícitas para controle das informações, conforme página oficial da biblioteca no Github ⁶.

4. Testes e Resultados

Para avaliação do desempenho dos algoritmos foi executada a segmentação das imagens do conjunto de validação da base de dados BSD500 (*Berkeley Segmentation Data Set*). O conjunto possui 100 imagens naturais com seus respectivos *ground truths* feito por anotações humanas [Arbelaez et al. 2011]. A classificação dos algoritmos foi feita utilizando o benchmark da base de dados BSDS500, conforme descrito na seção 3.3.

A fim de obter a melhor parametrização possível, foram executados diversos testes para cada algoritmo. Para o algoritmo EGB foi utilizada a parametrização informada pelos autores no artigo *Efficient Graph-Based Image Segmentation* [Felzenszwalb and Huttenlocher 2004]. A parametrização dos algoritmos está detalhada na Tabela 1.

Os algoritmos hierárquicos foram executados e um conjunto de partições foram selecionadas empiricamente para avaliação. O benchmark da base de dados BSDS500 retornou os melhores valores para cada um dos algoritmos, em duas métricas diferentes:

- ODS (*Optimal Data Set Scale*) - melhor parametrização para a base de dados [Arbelaez et al. 2009];
- OIS (*Optimal Image Scale*) - melhor parametrização para uma única imagem da base de dados [Arbelaez et al. 2009].

Os resultados da métrica *ODS* estão disponíveis nas Figuras 6 e 7. Além dos algoritmos avaliados, também está disponível um valor relativo ao desempenho de um “Humano” para comparação.

Conforme observamos na Figura 6, os algoritmos SGB e HSGB tiveram valores mais altos na métrica precisão, em relação ao algoritmo EGB e sua versão hierárquica.

⁴<https://github.com/scikit-image/scikit-image/tree/master/skimage/segmentation>

⁵<https://docs.scipy.org/doc/scipy-0.18.1/reference/generated/scipy.cluster.hierarchy.linkage.html>

⁶<https://github.com/scipy/scipy/tree/master/scipy/cluster>

Precisão x Revocação

Métodos de Segmentação de Imagens

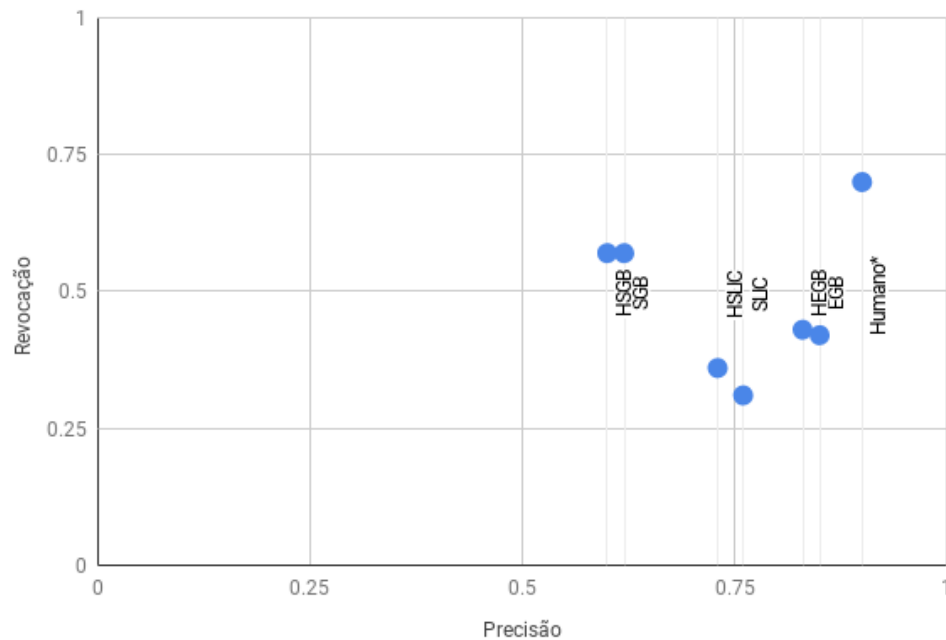


Figura 6. Valor médio de F -measure para os algoritmos avaliados.

Entretanto, o algoritmos baseados em EGB tiveram valores superiores na métrica de revocação. Comparando ambas as métricas, por meio da F -measure, observamos na Figura 6, que o algoritmo SGB teve valor ligeiramente superior ao algoritmo EGB. Ambos os algoritmos, e suas versões hierárquias, tiveram resultados consideravelmente superiores aos algoritmos SLIC.

A análise da Figura 6 indica que os algoritmos hierárquicos resultaram em valores superiores aos algoritmos sem hierarquização. Somente para o algoritmo HSGB, o resultado foi ligeiramente inferior ao obtido pelo algoritmo sem hierarquia. Em relação a variação dos resultados, a Figura 7 indica a variação entre os resultados obtidos pelos métodos. Podemos verificar que a variação existente para os algoritmos SGB, HSGB e HEGB é consideravelmente inferior aos demais algoritmos. Esse critério indica que esses algoritmos são mais consistentes, produzindo resultados semelhantes para diferentes imagens.

A partir dos resultados obtidos, verifica-se que algoritmo SLIC, apesar das vantagens de produzir superpixels de tamanhos regulares, apresentou a menor eficiência para segmentação de imagens, em relação aos demais algoritmos estudados. A combinação dos algoritmos SGB para detecção de contornos não apresentou resultado superior ao algoritmo EGB. Além disso, o custo computacional para execução dos dois algoritmos em sequência, bem como os custos para recoloração da imagens mostraram-se injustificáveis frente aos ganhos obtidos.

F-Measure

Métodos de Segmentação de Imagens

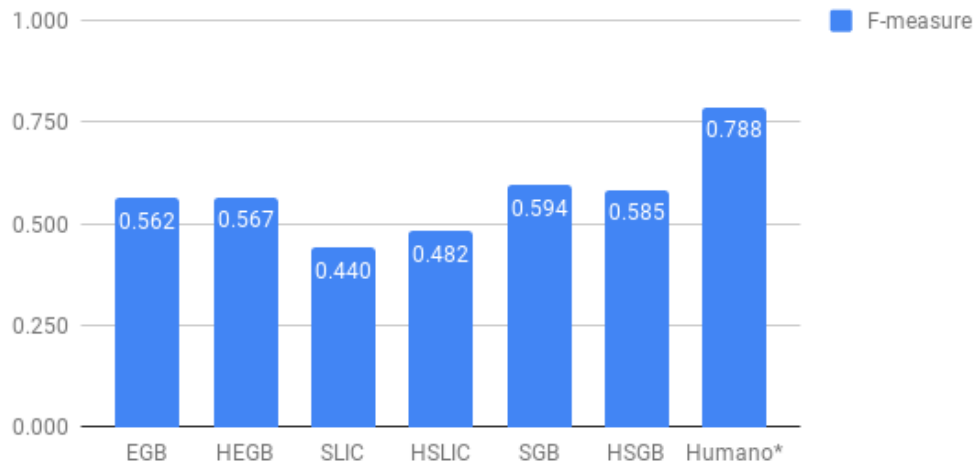


Figura 7. Valor máximo de *F-measure* para os algoritmos avaliados.

Threshold (Variação)

Métodos de Segmentação de Imagens

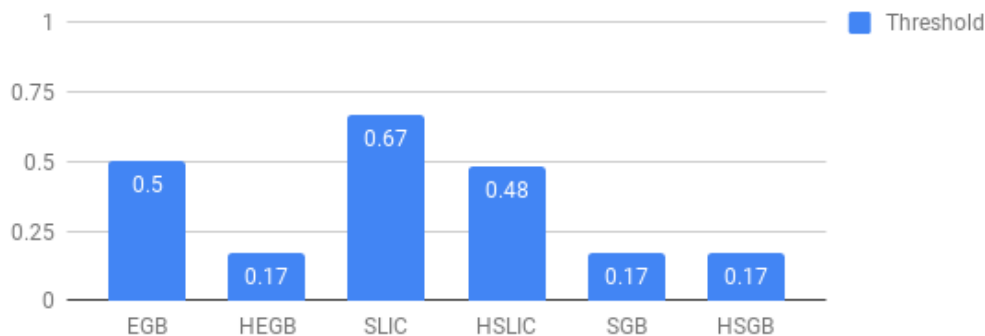


Figura 8. Valor máximo de *F-measure* para os algoritmos avaliados.

5. Conclusão

Os algoritmos de geração de superpixels avaliados nesse trabalho tiveram razoáveis para tarefas de segmentação e detecção de contornos, mesmo quando combinados. As versões hierárquicas dos algoritmos apresentaram resultados ligeiramente superiores àqueles obtidos sem hierarquias.

A utilização de hierarquias após a aplicação dos métodos de geração de *superpixels* aumentou consideravelmente os resultados de alguns métodos. Apesar da utilização não resultar em melhora significativa da capacidade de segmentação, justifica-se a utilização das versões hierárquicas em relação às versões originais dos algoritmos quando houver necessidade ampliação e redução de contornos, para efeito de agrupamento de informações. Os parâmetros descritos nesse trabalho podem ser utilizados como passo inicial para refinamento dos parâmetros em trabalhos futuros.

Os métodos hierárquicos apresentados no trabalho também permitem a utilização

em diferentes cenários, principalmente para redução e agrupamento de informação relevante em imagens, atendendo aos preceitos de análise multiescala.

Como trabalho futuro sugere-se a utilização dos algoritmos mostrados nesse trabalho como passo de pré processamento de outras técnicas de segmentação, como redes neurais convolucionais. Esses passos de segmentação iniciais podem permitir treinamento mais rápido e menor quantidade de informação a ser processadas, além de identificação mais rápida de bordas e contornos. Além disso, podem ser utilizados outros métodos de hierarquização, como o *complete-linkage*. Por fim, outros métodos de geração de superpixels podem ser comparados aos valores descritos nesse artigo, possibilitando comparação de diferentes métodos.

Referências

- Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, S. (2010). Slic superpixels. page 15.
- Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, S. (2012). Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282.
- Arbelaez, P. (2006). Boundary extraction in natural images using ultrametric contour maps. In *2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPRW'06)*, pages 182–182.
- Arbelaez, P., Maire, M., Fowlkes, C., and Malik, J. (2009). From contours to regions: An empirical evaluation. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2294–2301.
- Arbelaez, P., Maire, M., Fowlkes, C., and Malik, J. (2011). Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):898–916.
- Davis, J. and Goadrich, M. (2006). The relationship between precision-recall and roc curves. In *Proceedings of the 23rd International Conference on Machine Learning, ICML '06*, pages 233–240, New York, NY, USA. ACM.
- Dembczyński, K., Waegeman, W., Cheng, W., and Hüllermeier, E. (2011). An exact algorithm for f-measure maximization. In *Proceedings of the 24th International Conference on Neural Information Processing Systems, NIPS'11*, pages 1404–1412, USA. Curran Associates Inc.
- Domínguez, D. and Morales, R. R. (2016). *Image Segmentation: Advances*, volume 1.
- Felzenszwalb, P. F. and Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181.
- Fulkerson, B., Vedaldi, A., and Soatto, S. (2009). Class segmentation and object localization with superpixel neighborhoods. In *2009 IEEE 12th International Conference on Computer Vision*, pages 670–677.
- Guimarães, S. J. F. and Patrocínio, Z. K. G. (2013). A graph-based hierarchical image segmentation method based on a statistical merging predicate. In Petrosino, A., editor, *Image Analysis and Processing – ICIAP 2013*, pages 11–20, Berlin, Heidelberg. Springer Berlin Heidelberg.

- Jeyalakshmi, K. and Shanmugapriya, S. (2016). An efficient hierarchical clustering algorithms approach based on various-widths algometric clustering. In *International Journal of Innovative Research in Computer and Communication Engineering*, volume 4, pages 13151–13155.
- Liu, T., Seyedhosseini, M., and Tasdizen, T. (2016). Image segmentation using hierarchical merge tree. *IEEE Transactions on Image Processing*, 25(10):4596–4607.
- Murtagh, F. and Contreras, P. (2012). Algorithms for hierarchical clustering: An overview. 2:86–97.
- Pont-Tuset, J. and Marques, F. (2016). Supervised evaluation of image segmentation and object proposal techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 38(7):1465–1478.
- Rokach, L. and Maimon, O. (2005). *Clustering Methods*, pages 321–352. Springer US, Boston, MA.
- Sibson, R. (1973). Slink: An optimally efficient algorithm for the single-link cluster method. *Comput. J.*, 16:30–34.
- Torgo, L. and Ribeiro, R. (2009). Precision and recall for regression. In Gama, J., Costa, V. S., Jorge, A. M., and Brazdil, P. B., editors, *Discovery Science*, pages 332–346, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Vincent, L. and Soille, P. (1991). Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(6):583–598.
- Wang, M., Liu, X., Gao, Y., Ma, X., and Soomro, N. Q. (2017a). Superpixel segmentation: A benchmark. *Signal Processing: Image Communication*, 56:28 – 39.
- Wang, X.-Y., Wu, C.-W., Xiang, K., and Chen, W. (2017b). Efficient local and global contour detection based on superpixels. *Journal of Visual Communication and Image Representation*, 48:77 – 87.
- Zhang, H., Fritts, J. E., and Goldman, S. A. (2008). Image segmentation evaluation: A survey of unsupervised methods. *Computer Vision and Image Understanding*, 110(2):260 – 280.