





Segmentação de Imagens Utilizando a Abordagem de Grafos: Árvore Geradora Mínima e Caminho Mínimo


André Penchel   [Pontifical Catholic University of Minas Gerais | andre.penchel@sga.pucminas.br]

Túlio Gomes  [Pontifical Catholic University of Minas Gerais | tulio.gomes.1441272@sga.pucminas.br]

Luiz Starling  [Pontifical Catholic University of Minas Gerais | lastoledo@sga.pucminas.br]

Edson Pimenta  [Pontifical Catholic University of Minas Gerais | edson.almeida.1435541@sga.pucminas.br]

Gustavo Vinícius  [Pontifical Catholic University of Minas Gerais | gustavo.silva.1445251@sga.pucminas.br]

 Pontifical Catholic University of Minas Gerais, R. Dom José Gaspar, 500 - Coração Eucarístico, Belo Horizonte - MG, 30535-901, Brazil.

Resumo

Este trabalho implementa e analisa comparativamente dois métodos de segmentação de imagens baseados em teoria dos grafos: (i) a abordagem por Árvore Geradora Mínima (AGM) proposta por Felzenszwalb e Huttenlocher, e (ii) o método de Caminho Mínimo através da Transformação de Floresta de Imagens (IFT) desenvolvido por Falcão, Stolfi e Lotufo. A implementação em C++ aborda tanto imagens em níveis de cinza quanto coloridas, avaliando o desempenho dos algoritmos em termos de qualidade de segmentação e eficiência computacional. Os resultados demonstram as características distintivas de cada método: enquanto a AGM oferece segmentação automática e uniforme, a abordagem por caminhos mínimos proporciona maior controle através da seleção de sementes e melhor precisão em bordas complexas. A análise comparativa revela as vantagens e limitações de cada técnica em diferentes cenários, contribuindo para a compreensão prática desses importantes paradigmas de segmentação baseada em grafos.

Keywords: Segmentação de Imagens, Teoria dos Grafos, Árvore Geradora Mínima, Transformação de Floresta de Imagens, Algoritmo de Caminho Mínimo, Processamento Digital de Imagens, Implementação em C++, Análise Comparativa

1 Introdução

A segmentação de imagens é uma etapa fundamental no processamento digital de imagens, utilizada para dividir uma imagem em regiões com características semelhantes, como cor, textura ou intensidade. Neste trabalho, exploramos duas abordagens baseadas em grafos para realizar essa segmentação: uma por Árvore Geradora Mínima (MST), proposta por Felzenszwalb e Huttenlocher (2004), e outra por Caminho Mínimo, utilizando a Image Foresting Transform (IFT), desenvolvida por Falcão, Stolfi e Lotufo (2004). Ambas foram implementadas em C++, testadas em imagens em tons de cinza e coloridas, e comparadas quanto à eficiência e qualidade da segmentação gerada.

2 Funcionamento do Algoritmo de Segmentação Baseado em Árvore Geradora Mínima

O algoritmo proposto por Pedro F. Felzenszwalb e Daniel P. Huttenlocher tem como objetivo segmentar uma imagem em regiões perceptualmente coerentes. A abordagem é baseada em representar a imagem como um grafo e aplicar um critério adaptativo para decidir se duas regiões devem ser unidas ou não. A seguir, explicamos as principais etapas e conceitos envolvidos.

Representação da Imagem como Grafo

A imagem é representada como um grafo não direcionado $G = (V, E)$, em que:

- Cada vértice $v_i \in V$ representa um pixel da imagem;
- Cada aresta $(v_i, v_j) \in E$ conecta pixels vizinhos (ex: vizinhança 8);
- Cada aresta possui um peso $w((v_i, v_j))$ que indica a dissimilaridade entre os pixels, geralmente baseada na diferença de intensidade ou cor.

Medições de Diferença e Variação Interna

Diferença entre Componentes: Define-se a diferença entre duas regiões (ou componentes) C_1 e C_2 como o menor peso de aresta entre elas:

$$\text{Dif}(C_1, C_2) = \min_{\substack{v_i \in C_1, v_j \in C_2 \\ (v_i, v_j) \in E}} w((v_i, v_j))$$

Variação Interna: É a maior aresta da árvore geradora mínima (MST) de uma região C :

$$\text{Int}(C) = \max_{e \in \text{MST}(C)} w(e)$$

Cr terio de Decis o para Unir Regi es

A decis o de unir duas regi es   feita com base na compara  o entre a diferen a inter-regi es e a m nima varia  o interna aumentada:

$$\text{MInt}(C_1, C_2) = \min(\text{Int}(C_1) + \tau(C_1), \text{Int}(C_2) + \tau(C_2))$$

Onde a fun  o de limiar $\tau(C)$ penaliza componentes pequenas:

$$\tau(C) = \frac{k}{|C|}$$

O par metro k define a escala da segmenta  o: valores maiores tendem a gerar menos regi es (mais unificadas).

Predicado de Compara  o:

$$D(C_1, C_2) = \begin{cases} \text{true}, & \text{se } \text{Dif}(C_1, C_2) > \text{MInt}(C_1, C_2) \\ \text{false}, & \text{caso contr rio} \end{cases}$$

O Algoritmo

O algoritmo   inspirado no de Kruskal e segue os passos abaixo:

1. Ordenar todas as arestas por peso crescente;
2. Inicializar cada pixel como uma regi o separada;
3. Iterar pelas arestas ordenadas:
 - Se os v rtices da aresta pertencem a componentes diferentes e o predicado D retornar false, une-se as duas regi es.
4. Repetir at  percorrer todas as arestas.

A estrutura de dados utilizada   a de conjuntos disjuntos (Union-Find) com compress o de caminho e uni o por rank, garantindo complexidade eficiente: $O(m \log m)$, onde m   o n mero de arestas.

Propriedades Importantes

Apesar de tomar decis es locais (guloso), o algoritmo garante que a segmenta  o final:

- N o seja muito fina nenhuma regi o separada poderia ser unida sem violar o predicado;
- Nem muito grosseira se houver evid ncia clara de uma fronteira, ela ser  respeitada.

Par metros e Aplica  es

O par metro principal do algoritmo   o valor de k , que controla o grau de detalhe:

- k pequeno: segmenta  es mais detalhadas;
- k grande: regi es maiores e menos segmentadas.

O algoritmo funciona bem tanto com grafos baseados em vizinhan a na grade (grid) quanto em grafos de vizinhos mais pr ximos no espa o de atributos (ex: posi  o + cor). Al m disso, pode ser aplicado a imagens coloridas, rodando separadamente nos canais R, G e B e depois cruzando os resultados.

3 Algoritmo de Transforma  o de Floresta de Imagens (IFT)

A Transforma  o de Floresta de Imagens (Image Foresting Transform - IFT), proposta por [?],   uma t cnica de segmenta  o que v  a imagem como um grafo e resolve o problema por meio da teoria de caminhos m nimos. A ideia principal   construir uma floresta de caminhos  timos, onde cada pixel est  conectado a uma semente (ponto inicial escolhido) atrav s do melhor caminho poss vel, segundo algum cr terio de custo.

3.1 Representa  o da Imagem como Grafo

Neste m todo, a imagem   tratada como um grafo direcionado $G = (V, E)$, onde:

- Cada pixel da imagem   um v rtice $v_i \in V$
- Os arcos $(v_i, v_j) \in E$ conectam pixels vizinhos de acordo com uma rela  o de adj c ncia (por exemplo, 4 ou 8 vizinhos)
- Cada arco possui um custo $w(v_i, v_j)$ que mede a dificuldade de ir de v_i para v_j . Esse custo pode ser baseado em cor, gradiente ou outras caracter sticas locais da imagem

3.2 Fun  es de Custo de Caminho

Para decidir qual caminho entre um pixel e uma semente   melhor, usamos fun  es de custo. As duas principais s o:

3.2.1 Custo Aditivo

Essa fun  o soma os custos de todos os arcos no caminho:

$$f_{\text{sum}}(\pi \cdot \langle s, t \rangle) = f_{\text{sum}}(\pi) + w(s, t) \quad (1)$$

Ou seja, o custo de um novo caminho   o custo do caminho anterior mais o custo do novo arco adicionado.

3.2.2 Custo M ximo de Arco

Essa fun  o considera o arco mais caro no caminho:

$$f_{\text{max}}(\pi \cdot \langle s, t \rangle) = \max\{f_{\text{max}}(\pi), w(s, t)\} \quad (2)$$

 til para aplica  es onde o maior obst culo no caminho   o que mais importa (ex: bacias hidrogr ficas).

3.3 O que o Algoritmo Produz

O algoritmo constr i tr s estruturas:

1. Predecessores $P(t)$: guarda de onde veio cada pixel no caminho
2. Custos $C(t)$: o custo total para alcan ar cada pixel a partir de uma semente
3. R tulos $L(t)$: indica a qual semente cada pixel est  conectado ( til para segmenta  o)

Esses caminhos são ótimos porque para cada pixel t , temos:

$$f(P^*(t)) = \min_{\forall \pi \text{ terminando em } t} f(\pi) \quad (3)$$

Ou seja, o algoritmo garante o menor custo possível até cada ponto da imagem.

3.4 Passo a Passo do Algoritmo IFT

O algoritmo Image Foresting Transform (IFT) recebe como entrada uma imagem I , um conjunto de sementes \mathcal{S} e uma função de custo f . Seu objetivo é construir uma floresta de caminhos ótimos que conecta cada pixel da imagem a uma semente, de forma que o custo total do caminho seja minimizado. O algoritmo gera três estruturas principais: a floresta de predecessores P , os custos mínimos C e os rótulos L .

1. Inicialmente, para cada pixel t da imagem, define-se o custo $C(t)$ como $f(\langle t \rangle)$, ou seja, o custo de iniciar um caminho no próprio pixel. O predecessor $P(t)$ é definido como nulo, indicando que ainda não há caminho formado.
2. Todos os pixels pertencentes ao conjunto de sementes \mathcal{S} são inseridos em uma fila de prioridade \mathcal{Q} . Esses pixels representam os pontos de partida da floresta e recebem rótulos iniciais distintos.
3. Enquanto a fila de prioridade \mathcal{Q} não estiver vazia, o seguinte processo é repetido:
 - Remove-se da fila o pixel s com menor custo atual.
 - Para cada vizinho t de s , calcula-se o custo C' de alcançar t por meio de s . Esse custo é dado por $C' = f(P^*(s) \cdot \langle s, t \rangle)$, onde $P^*(s)$ é o caminho ótimo até s e $\langle s, t \rangle$ é a aresta entre os dois pixels.
 - Se $C' < C(t)$, significa que foi encontrado um caminho melhor para t . Nesse caso, atualiza-se:
 - O predecessor: $P(t) \leftarrow s$
 - O custo: $C(t) \leftarrow C'$
 - O rótulo: $L(t) \leftarrow L(s)$
 - A posição de t na fila de prioridade é atualizada com o novo custo.
4. O algoritmo termina quando todos os pixels tiverem sido processados, ou seja, quando a fila estiver vazia. Ao final, a floresta de caminhos mínimos estará construída, segmentando a imagem conforme os critérios definidos pela função de custo.

4 Correlações

Os métodos de Árvore Geradora Mínima (AGM) e Transformação de Floresta de Imagens (IFT) compartilham fundamentos teóricos em teoria dos grafos, mas diferem em estratégias e aplicações. Esta seção explora suas relações conceituais e práticas.

4.1 Elementos Comuns

- Representação como grafo: Ambos modelam a imagem como um grafo $G = (V, E)$ onde:
 - Vértices representam pixels
 - Arestas conectam vizinhos (4/8-conexos ou outros critérios)
- Critérios baseados em pesos: Utilizam funções de dissimilaridade $w(v_i, v_j)$ baseadas em:

Cor: $\|I(v_i) - I(v_j)\|$ Gradiente: $\|\nabla I(v_i) - \nabla I(v_j)\|$
- Abordagem gulosa: Ambos constroem a solução ótima através de decisões locais

4.2 Divergências Fundamentais

A comparação entre os métodos AGM e IFT revela diferenças fundamentais em sua estrutura e funcionamento. O algoritmo AGM utiliza um grafo não-direcionado, no qual a estrutura resultante é uma floresta composta por árvores geradoras mínimas. Seu controle de segmentação se dá por meio do parâmetro k , que ajusta o nível de detalhe da segmentação. O critério que define se regiões devem ser unidas é baseado na diferença entre regiões adjacentes, levando em consideração a variação interna de cada componente. Sua complexidade computacional é $O(m \log m)$, onde m representa o número de arestas do grafo.

Por outro lado, o método IFT opera sobre um grafo direcionado, no qual a estrutura gerada é uma floresta de caminhos ótimos a partir de sementes pré-definidas. Nesse caso, o controle do usuário ocorre através da escolha dessas sementes, o que permite uma segmentação mais guiada e precisa. O critério de conexão entre os pixels baseia-se no custo acumulado do caminho, considerando medidas como diferença de intensidade, gradientes ou distância. A complexidade do algoritmo é $O(m + n \log n)$, sendo n o número de nós (pixels) e m o número de conexões (arestas) no grafo.

4.3 Complementaridade na Segmentação

- AGM: Excelente para segmentação automática quando:
 - As regiões têm características homogêneas internas
 - Não há conhecimento prévio sobre objetos de interesse
- IFT: Mais adequado quando:
 - Existem sementes/indicadores das regiões relevantes
 - Requer-se precisão em bordas complexas (ex: gradientes fracos)

4.4 Integração Potencial

Ambos métodos podem ser combinados em pipelines de processamento:

1. Usar AGM para segmentação inicial e detecção de regiões candidatas
2. Selecionar sementes automaticamente nos centróides das regiões da AGM
3. Aplicar IFT com sementes refinadas para delimitação precisa de bordas

4.5 Análise Quantitativa

Principais métricas comparativas:

$$\text{Sensibilidade a ruído} = \begin{cases} \frac{1}{k} & (\text{AGM}) \\ \frac{1}{|\mathcal{S}|} & (\text{IFT}) \end{cases} \quad (4)$$

$$\text{Uniformidade regional} = \frac{\sum_{R_i \in \mathcal{R}} \text{Var}(R_i)}{|\mathcal{R}|} \quad (5)$$

onde \mathcal{R} é o conjunto de regiões segmentadas.

4.6 Casos de Uso Recomendados

- Prefira AGM para:
 - Segmentação automática de texturas
 - Cenas com iluminação uniforme
- Prefira IFT para:
 - Segmentação interativa com sementes
 - Imagens médicas com marcadores anatômicos

5 Resultados obtidos

5.1 AGM

Nesta subseção, são apresentados os resultados da segmentação utilizando o algoritmo de Felzenszwalb e Huttenlocher (AGM), com foco na comparação de duas imagens distintas processadas sob os mesmos parâmetros. Foi adotado um valor de $\sigma = 1.0$ para o filtro Gaussiano, responsável pela suavização da imagem antes da construção do grafo, e um valor de $k = 1000$, que controla o critério de diferença interna utilizado na decisão de fusão dos componentes.

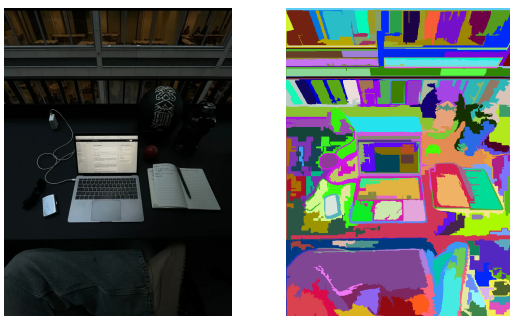


Figura 1. $\sigma = 1.0$ e $k = 1000$

5.2 IFT

A Figura 2 apresenta os resultados da segmentação utilizando o algoritmo de Caminho Mínimo (IFT) aplicando diferentes níveis de similaridade: 5%, 10% e 20%. Quanto maior o nível de similaridade adotado ou seja, um limiar mais baixo, menor é a tolerância às variações internas da imagem. Isso gera uma segmentação mais sensível às pequenas diferenças locais, resultando em mais detalhes nas fronteiras. Com limiares mais altos, as áreas são agrupadas de forma mais ampla, gerando menos divisões e uma segmentação mais suave.

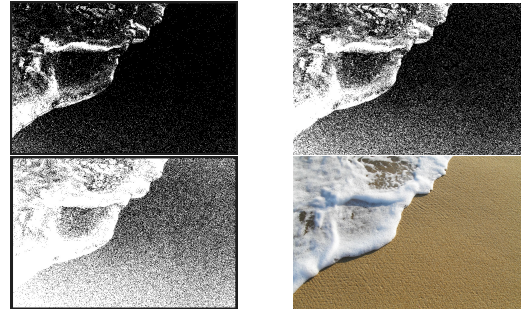


Figura 2. Segmentação utilizando o IFT com diferentes níveis de similaridade: 5%, 10% e 20%.

6 Divisão de Tarefas entre os Integrantes

- André Penchel: Leitura e resumo do artigo da AGM, implementação do predicado adaptativo, estruturação do relatório em \LaTeX .
- Túlio Braga: Leitura do artigo do IFT, adaptação da propagação em C++, testes em imagens em tons de cinza.
- Edson Pimenta: Análise comparativa dos métodos e organização dos resultados.
- Luis Starling: Implementação do algoritmo MST em C++, testes em imagens coloridas.
- Gustavo Vinícius: Escrita da seção de correlação entre os métodos, experimentação e organização dos gráficos/imagens.

Referências

- [1] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision (IJCV)*, vol. 59, no. 2, pp. 167–181, 2004.
- [2] Alexandre. X. Falcão, Jorge Stolfi and Roberto de Alencar Lotufo, The image foresting transform: theory, algorithms, and applications, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 19-29, Jan. 2004, doi: 10.1109/TPAMI.2004.1261076.