

Segmentação de Imagens Baseada em Grafos: Comparação entre os Algoritmos de Felzenszwalb e Huttenlocher e de Boykov e Funka-Lea

Arthur Gonçalves de Moraes

15 de Dezembro de 2024

Resumo

A segmentação de imagens é uma etapa fundamental em diversas aplicações de visão computacional. Este artigo apresenta uma análise detalhada de dois algoritmos de segmentação baseados em grafos: o algoritmo eficiente de Felzenszwalb e Huttenlocher (2004) e o método de cortes em grafos s-t de Boykov e Funka-Lea (2006). Ambos os algoritmos utilizam grafos para representar imagens e realizar segmentações, mas diferem significativamente em termos de objetivos, abordagem teórica e eficiência computacional. Discutimos os aspectos teóricos e práticos de cada implementação, ilustramos resultados com diferentes configurações de parâmetros e avaliamos a qualidade das segmentações geradas.

1 Introdução

A segmentação de imagens visa dividir uma imagem em regiões que compartilhem características semelhantes, como cor, textura ou intensidade. Algoritmos eficientes e robustos são essenciais para aplicações em reconhecimento de padrões, rastreamento de objetos e compressão de imagens. Diversos métodos baseados em grafos foram desenvolvidos para atender a esses objetivos.

O trabalho seminal de Felzenszwalb e Huttenlocher (2004) introduziu um algoritmo que utiliza uma abordagem ávida e eficiente para segmentar imagens em componentes conexos, com custo computacional de $O(n \log n)$. Por outro lado, o algoritmo de Boykov e Funka-Lea (2006) adota uma abordagem baseada em cortes de grafos s-t, buscando otimizar globalmente uma função de energia, com flexibilidade para aplicações em dados N-dimensionais.

Este artigo aborda os dois algoritmos, comparando suas abordagens, implementações e aplicações.

2 Descrição dos Algoritmos

2.1 Felzenszwalb e Huttenlocher (2004)

O algoritmo de Felzenszwalb e Huttenlocher representa uma imagem como um grafo $G = (V, E)$, onde cada pixel é um vértice $v \in V$, e as arestas $e \in E$ conectam vértices vizinhos com pesos baseados na dissimilaridade. As etapas principais incluem:

- **Cálculo da Dissimilaridade:** A dissimilaridade entre dois pixels a e b é dada por $|I(a) - I(b)|$, onde I é a intensidade.
- **Organização por Componentes:** Utiliza-se um algoritmo de união e busca para gerenciar componentes conexos.
- **Critério de Segmentação:** Dois componentes C_1 e C_2 são unidos se a menor aresta entre eles for menor que um limiar ajustado por um parâmetro k , definido como $\tau(C) = k/|C|$, onde $|C|$ é o tamanho do componente.

Este método é particularmente eficiente para capturar regiões homogêneas em imagens e é aplicável em tempo real devido à sua complexidade reduzida.

2.2 Boykov e Funka-Lea (2006)

O método de Boykov e Funka-Lea adota uma abordagem de otimização global baseada em cortes mínimos em grafos. O grafo contém arestas que conectam pixels a dois terminais especiais, representando as classes "objeto" e "fundo".

- **Representação do Grafo:** Além das arestas de vizinhança (n-links), que conectam pixels vizinhos, são incluídas arestas terminais (t-links), que conectam pixels aos terminais.
- **Função de Energia:** A segmentação é obtida minimizando uma função de energia que equilibra termos de borda e região:

$$E(A) = \lambda \cdot R(A) + B(A),$$

onde $R(A)$ mede a compatibilidade com modelos regionais, e $B(A)$ penaliza descontinuidades nas bordas.

- **Minimização:** O corte mínimo no grafo é calculado usando algoritmos de fluxo máximo (e.g., Ford-Fulkerson), garantindo uma solução globalmente ótima.

Este método é flexível e pode incorporar restrições adicionais, como sementes para forçar pixels a pertencerem a uma determinada classe.

3 Implementação

3.1 Implementação do Algoritmo de Felzenszwalb e Huttenlocher

A implementação do algoritmo de Felzenszwalb e Huttenlocher foi realizada com os seguintes passos:

1. **Carregamento da Imagem:** Imagens em formato '.pgm' são convertidas para matrizes de intensidades.
2. **Construção do Grafo:** Arestas são criadas para os pixels vizinhos (4-conexões), com pesos baseados na diferença de intensidade.
3. **Organização por Componentes:** Um algoritmo de união e busca com compressão de caminho é usado para gerenciar componentes conexos.
4. **Critério de Parada:** O algoritmo para quando o limiar de dissimilaridade é atingido, determinado pelo parâmetro k .
5. **Geração da Saída:** Imagens segmentadas são geradas e salvas em formato '.ppm'.

3.2 Implementação do Algoritmo de Boykov e Funka-Lea

A implementação do algoritmo de Boykov e Funka-Lea foi realizada da seguinte forma:

1. **Carregamento da Imagem:** Imagens em formato '.pgm' são convertidas para matrizes de intensidades.
2. **Construção do Grafo:** São definidos dois terminais, a fonte e o sumidouro, e pesos são atribuídos às arestas (n-links e t-links) com base na função de energia.
3. **Cálculo do Corte Ótimo:** Algoritmo de fluxo máximo Edmonds-Karp (BFS-based Ford-Fulkerson) é utilizados para encontrar o corte mínimo que segmenta o grafo.
4. **Geração da Saída:** As imagens segmentadas são salvas em formato '.ppm'.

4 Estruturas de Dados Utilizadas

A implementação dos algoritmos de segmentação de imagens descritos neste artigo faz uso de estruturas de dados específicas para representar imagens e grafos. Estas estruturas são fundamentais para garantir a eficiência e a simplicidade das operações realizadas pelos algoritmos.

4.1 Representação da Imagem

As imagens são convertidas para matrizes 2D de inteiros, onde cada elemento corresponde à intensidade de um pixel. Esta representação permite um acesso eficiente às informações de cada pixel e é a base para a construção do grafo utilizado nos algoritmos de segmentação. Cada entrada na matriz representa a intensidade de cinza do pixel em imagens monocromáticas ou a média das intensidades em imagens coloridas.

4.2 Representação do Grafo

O grafo utilizado na segmentação é implementado por meio de uma estrutura que combina:

- **Lista de Adjacência:** Cada nó do grafo é associado a uma lista que armazena seus vizinhos diretos. Isso facilita a navegação eficiente entre os nós conectados.
- **Lista de Arestas:** As conexões entre os nós são armazenadas em uma estrutura que contém informações sobre as arestas, como os nós de origem e destino, a capacidade (peso) e o fluxo atual. Esta organização é particularmente importante para algoritmos que requerem atualizações dinâmicas, como o método de Boykov e Funka-Lea.

Para representar a conectividade entre os pixels da imagem, cada pixel é tratado como um nó do grafo, e as arestas conectam pixels vizinhos. Os pesos das arestas são calculados com base na dissimilaridade entre pixels, que pode ser definida pela diferença de intensidade ou por critérios adicionais, como gradientes ou textura.

Essa abordagem modular facilita a construção de grafos complexos e permite o uso eficiente de algoritmos de segmentação baseados em grafos. Além disso, a inclusão de arestas especiais, como as que conectam os pixels aos terminais de fonte e sumidouro no método de Boykov e Funka-Lea, garante que a estrutura seja adaptável a diferentes aplicações.

5 Resultados e Discussão

Os experimentos compararam os dois algoritmos em termos de precisão, granularidade da segmentação e tempo de execução. Os principais achados foram:

- O algoritmo de Felzenszwalb é mais rápido porém apresenta segmentações menos precisas.
- O método de Boykov oferece maior flexibilidade e precisão em segmentações complexas, mas é consideravelmente mais lento ($O(x^3)$ em nossa implementação).



Imagem original.



(a) Segmentação utilizando o algoritmo 1, com $K=25000$.



(b) Segmentação utilizando o algoritmo 2, com $\lambda=10.0$ e $\sigma=0.1$.

6 Conclusão

Ambos os algoritmos demonstram eficácia em seus contextos de aplicação. Enquanto o método de Felzenszwalb se destaca pela eficiência em segmentação rápida, o de Boykov oferece maior robustez e flexibilidade. Trabalhos futuros podem explorar a integração das vantagens de ambos os métodos para criar uma abordagem híbrida.

Referências

- [1] Felzenszwalb, P. F., & Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. *International Journal of Computer Vision*.
- [2] Boykov, Y., & Funka-Lea, G. (2006). Graph cuts and efficient N-D image segmentation. *International Journal of Computer Vision*.
- [3] Shi, J., & Malik, J. (1997). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.