



Diego Martin Mancini
Orientador Prof. Dr. Paulo A. V. de Miranda
Instituto de Matemática e Estatística da Universidade de São Paulo

Introdução

Com o advento de novas tecnologias de aquisição de imagens, a necessidade de gerenciar e manipular grandes volumes de informações de multimídia tem crescido nos últimos anos na internet. Para isso, são utilizados sistemas de recuperação de imagens baseados em conteúdo (Content-Based Image Retrieval - CBIR) que permitem a recuperação de imagens com base nas características de cor, textura e forma dos objetos presentes em uma imagem. O método usado na extração dessas características é chamado de "descritor". Um desses descritores, o BIC - que utiliza a extração de característica de cor -, propõe um método eficaz em termos de acurácia, classificando os pixels da imagem como interior e borda. No entanto, uma das etapas de processamento do BIC, que classifica o pixel em interior/borda, não utiliza algoritmos tradicionais da literatura computacional em processamento de imagem.

Objetivo

Levando em consideração o algoritmo utilizado para classificar os pixels em interior/borda no BIC, a proposta desse trabalho é substituí-lo por outros algoritmos considerados clássicos e que poderiam, em teoria, ser utilizados para a mesma finalidade, mas que não foram testados. Para esse trabalho, o BIC será analisado em conjunto com o detector de bordas de Canny e o Operador Laplaciano.

Sendo assim, o objetivo aqui é realizar uma comparação do descritor BIC convencional com suas outras versões utilizando os supracitados algoritmos e testar se, de fato, o BIC é tão eficaz (em termos de acurácia) quando comparado com suas variações que utilizam os algoritmos de detecção de bordas de Canny e o operador Laplaciano.

BIC

Desenvolvido por Stehling et. al. [1], esse descritor utiliza o espaço de cores RGB uniformemente quantizado em um número de cores, em geral $4 \times 4 \times 4 = 64$. Este descritor gera dois histogramas de cor, um para os pixels de borda e outro para os pixels de interior. Um pixel é considerado interior se seus 4 vizinhos tiverem a mesma cor (após a quantização da imagem), caso contrário será borda.

Em seguida, os dois histogramas de cores são concatenados resultando em um vetor que representa as características extraídas pelo BIC.

Algorithm 1 BIC

Entrada:

img é a imagem de tamanho $n \times m$.

Saída:

o histograma de características de cor binária (vetor).

função BIC(*img*)

prop ← *quantiza_pixels(img)* /*Quantifica as cores em 64 níveis (4x4x4).*/

prop ← *borda_interior(prop)* /*Classifica os pixels em interior e borda.*/

vetor ← *cria_histograma(prop)* /*Cria o histograma.*/

devolve *vetor*

fim função

Figura 1: Pseudocódigo do descritor BIC.

Depois de construído o vetor de características é necessário um método para compararmos histogramas de cor. Esse método é a distância dLog, representado pela equação abaixo.

$$dLog(q, d) = \sum_{i=0}^{i < M} [f(q[i]) - f(d[i])]$$

$$f(x) = \begin{cases} 0, & \text{se } x = 0 \\ 1, & \text{se } 0 < x \leq 1 \\ [\log_2 x] + 1 & \text{caso contrário} \end{cases}$$

Onde q e d são histogramas de cores com M cores cada.

Variações do BIC com detector de bordas de Canny e Operador Laplaciano

Seguindo o que foi proposto no trabalho, os algoritmos de Canny e Operador Laplaciano substituirão o algoritmo usual de detecção de interior/borda do BIC (Figura 1) para a etapa da extração de características de cor. Abaixo está a comparação de imagens obtidas pelo BIC com algoritmo convencional e pelas suas versões alternativas.

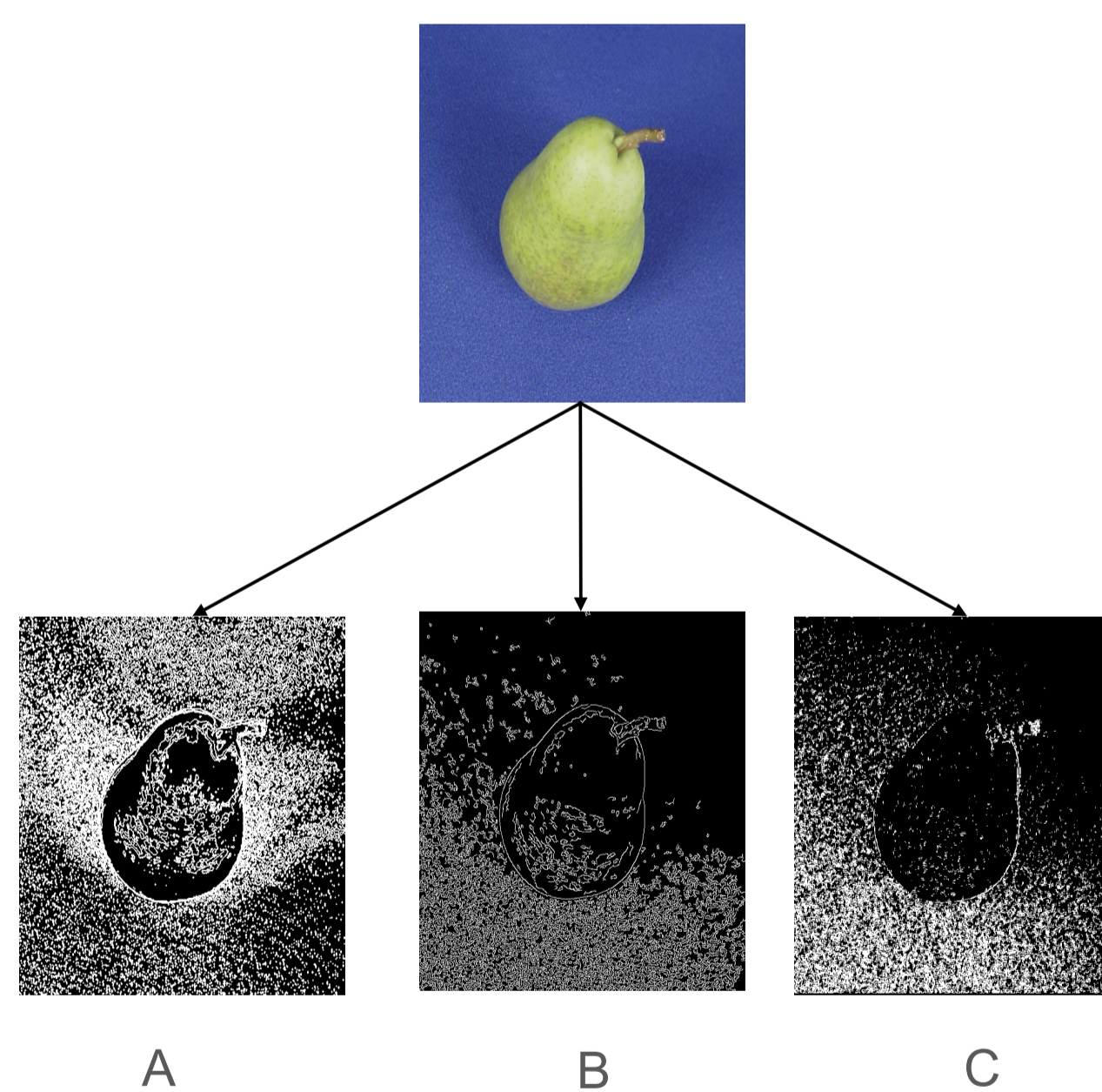


Figura 2: Comparação entre imagem gerada pelo BIC original (A), BIC com Canny (B) e BIC com Laplaciano (C).

Esses dois métodos alternativos utilizam uma janela (kernel) 3×3 e um limiar (threshold) de 20%.

O Canny [2] segue os seguintes três critérios: boa detecção, boa localização e a premissa de detecção única de borda. Esses critérios marcam todas as bordas possíveis da imagem uma única vez.

Já o operador Laplaciano utiliza o operador de Sobel [3] para classificar a intensidade de cada pixel que será usada para calcular as bordas com o zero crossing [4].

Experimento

Para a realização do experimento, foi necessário um conjunto de imagens devidamente selecionado com 54 classes, onde cada classe possui 41 imagens. Usando tais classes de imagens, os descritores BIC e suas variações com Canny e o Operador Laplaciano construirão os histogramas que serão utilizados na comparação entre as imagens usadas para o cálculo de distância dLog. Depois de calculada tal distância, os dados obtidos resultarão na criação dos gráficos de *precision x recall*. Dessa forma, ter-se-á maior clareza sobre a comparação das diferentes versões para o descritor BIC em termos de acurácia.

Resultados

Os resultados serão adquiridos a partir da seleção de alguns pontos do eixo *recall* e seus respectivos valores para o eixo *precision*, além da observação do gráfico como um todo. As três curvas de cada descritor dos gráficos resultaram ser bem semelhantes. Através dessa análise podemos concluir que, de maneira generalizada, os três métodos são semelhante em números absolutos, como pode ser observado no gráfico abaixo.

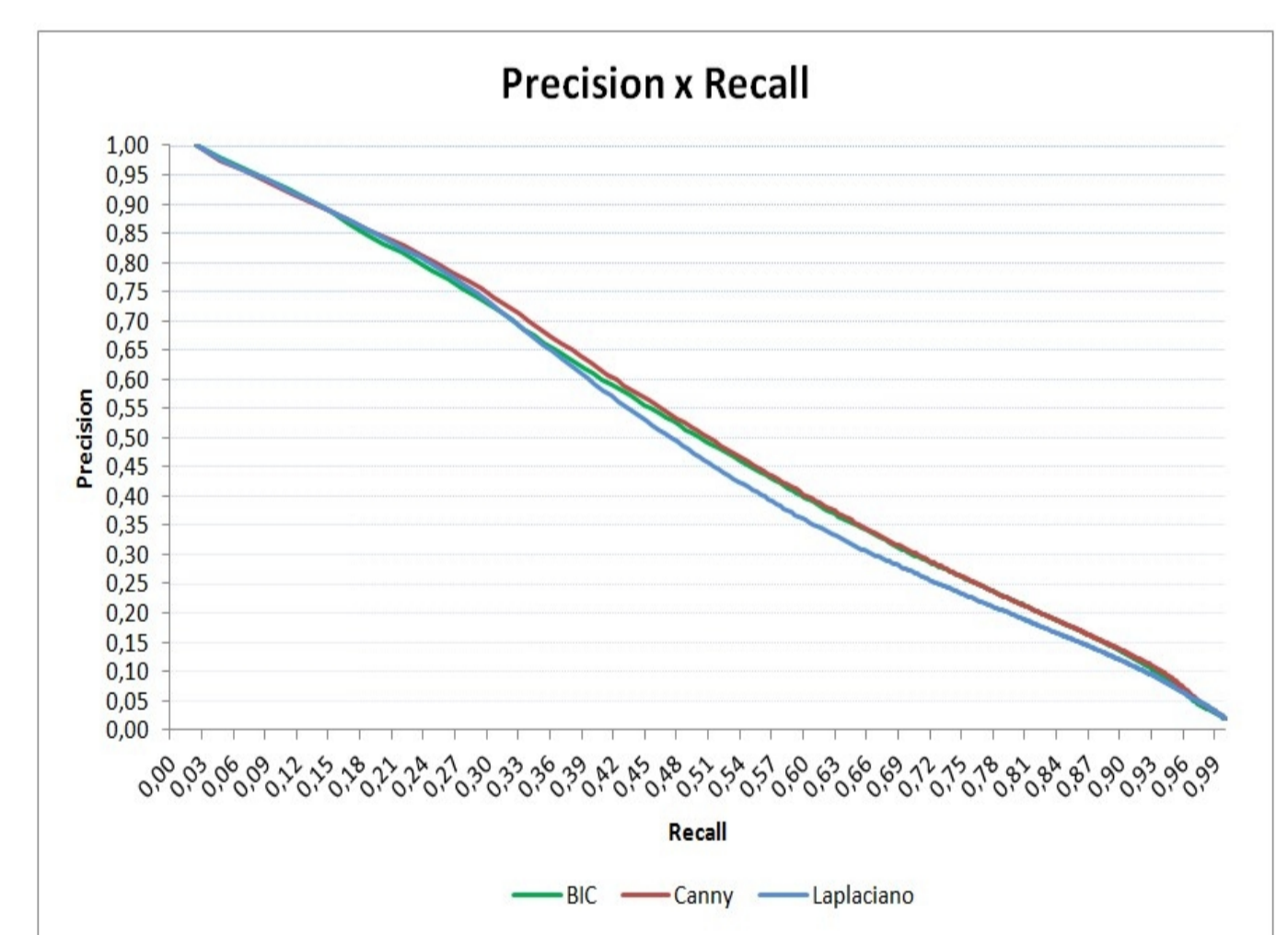


Figura 3: Gráfico da comparação do descritor BIC e suas variações

Conclusão

Através dos resultados obtidos após os testes realizados, podemos concluir de maneira satisfatória que o descritor BIC e suas variações com Canny e o Laplaciano possuem desempenhos semelhantes em termos de acurácia, ainda que o descritor BIC na sua versão com Laplaciano tenha apresentado uma menor acurácia em comparação com o BIC original e com a sua versão com Canny.

Referências Bibliográficas:

- [1] R. O. Stehling, M. A. Nascimento, and A. X. Falcão. A compact and efficient image retrieval approach based on border/interior pixel classification. CIKM '02 Proceedings of the eleventh international conference on Information and knowledge management, 2002. Wikipedia <https://en.wikipedia.org/wiki/Canny_edge_detector> Acessado em 08-2015. [3] R. F. Gonzalez and R. E. Woods. Digital image processing. Pearson PrenticeHall, 2008 [4] Zero crossing laplacian operator. <<http://fourier.eng.hmc.edu/e161/lectures/gradient/node7.html>> Acessado em 10-2015