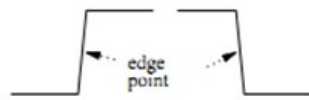


Danilo de Santana Pena

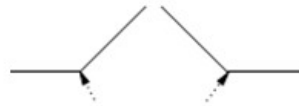
Natal, Novembro de 2012

Introdução

O objetivo deste trabalho é a extração de *features* através do uso de filtros, realçando as características de interesse. Uma característica importante encontrado nas imagens são as bordas, pois elas trazem o significado contorno presente na imagem. Abaixo segue algumas classificações de bordas:



a) Step



b) Concave slope



c) Convex slope



d) Roof



e) Valley



f) Staircase



g) Peak

Filtros

Alguns filtros têm por objetivo realçar as bordas, sabendo-se que as bordas são representadas por variações bruscas dos pixels de intensidade, ou seja, componentes de alta frequência no espectro. As máscaras e definições dos filtros podem ser encontradas no livro “Digital Image Processing” [2].

Alguns filtros espaciais foram implementados em C++ utilizando Qt e OpenCV, um programa foi desenvolvido com os principais filtros, como segue:

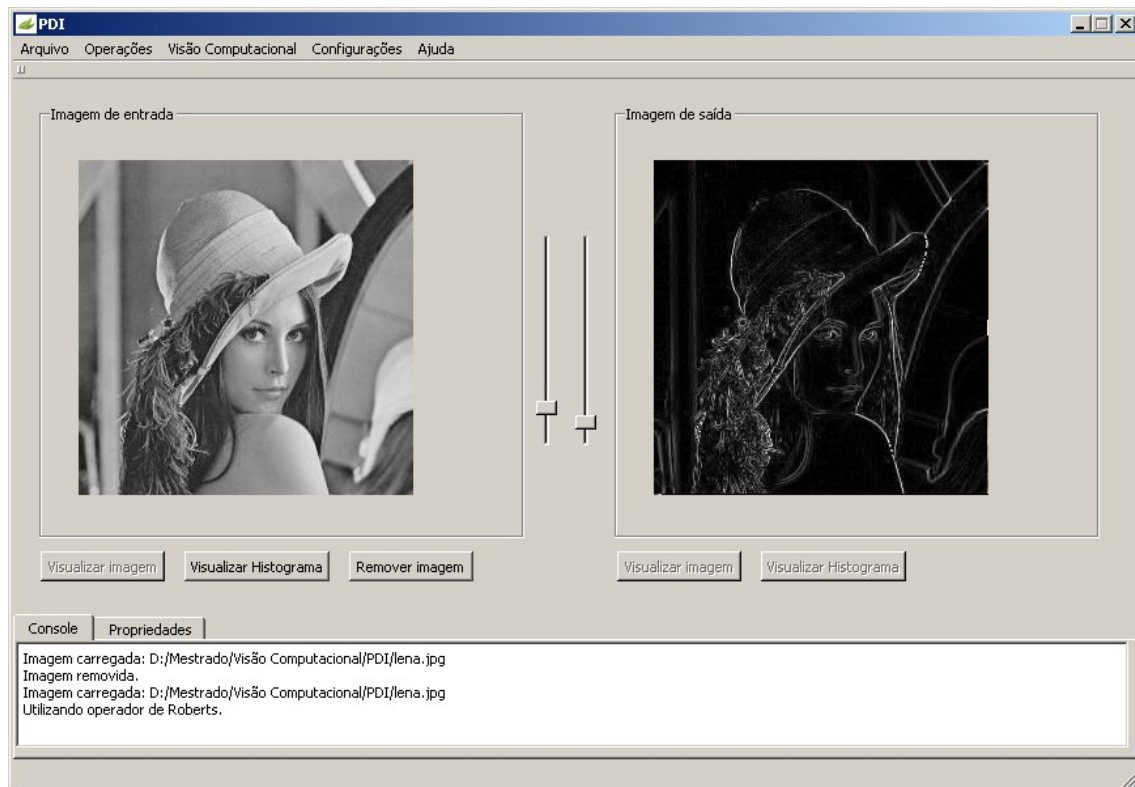


Figure 1: Programa desenvolvido.

A imagem de entrada é a imagem da Lena, com dimensões 256x256.



Figure 2: Lena.

Filtro de Roberts:

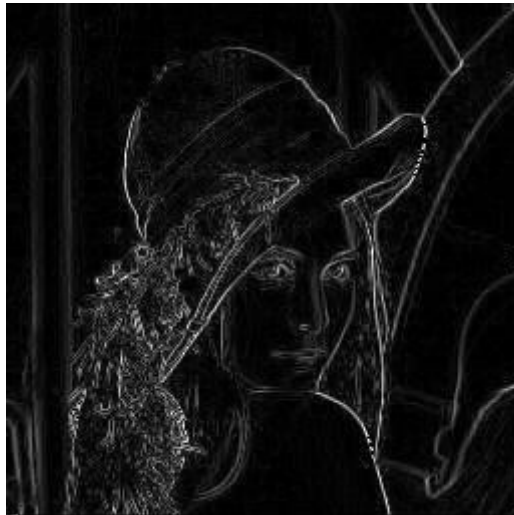


Figure 3: Resultado do filtro de Roberts.

Filtro de Sobel:



Figure 4: Resultado do filtro de Sobel.

Filtro Laplaciano:



Figure 5: Resultado do filtro do Laplaciano.

Método de Canny

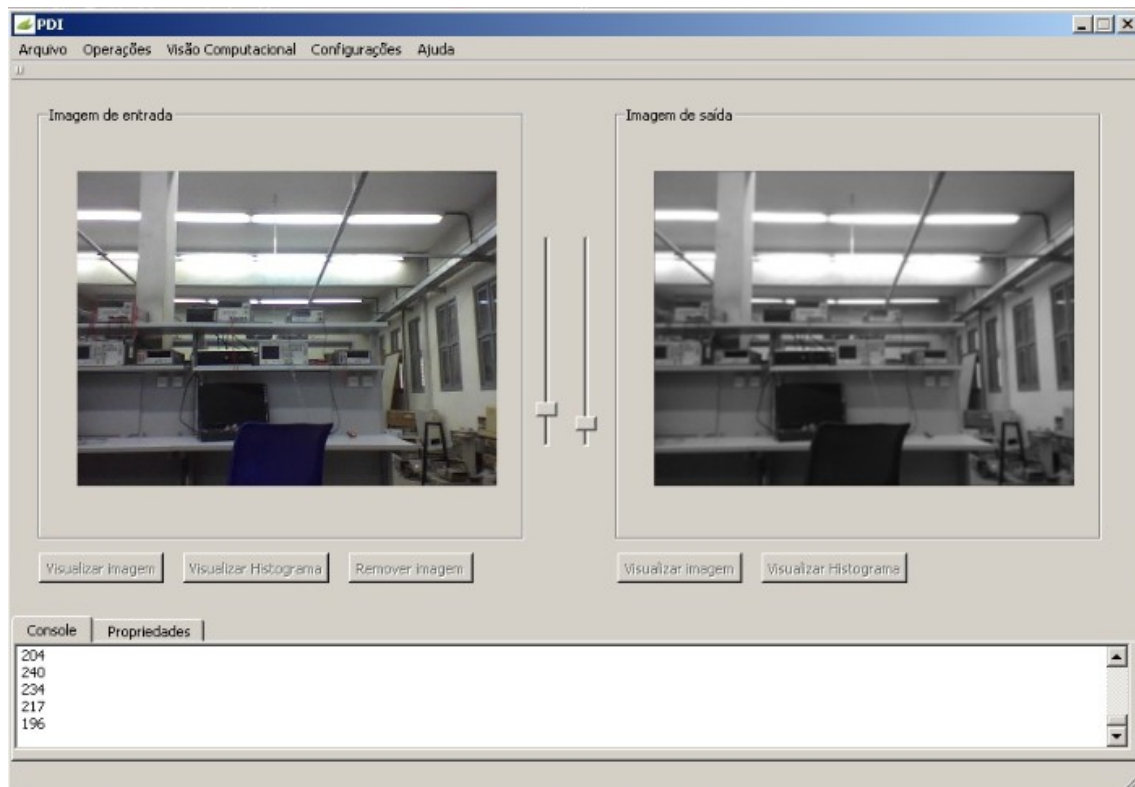
O operador de Canny foi proposto por John Canny em 1986 [1], com os seguintes critérios de desempenho: boa detecção, boa localização e um único resultado de borda.

As seguintes etapas foram implementadas:

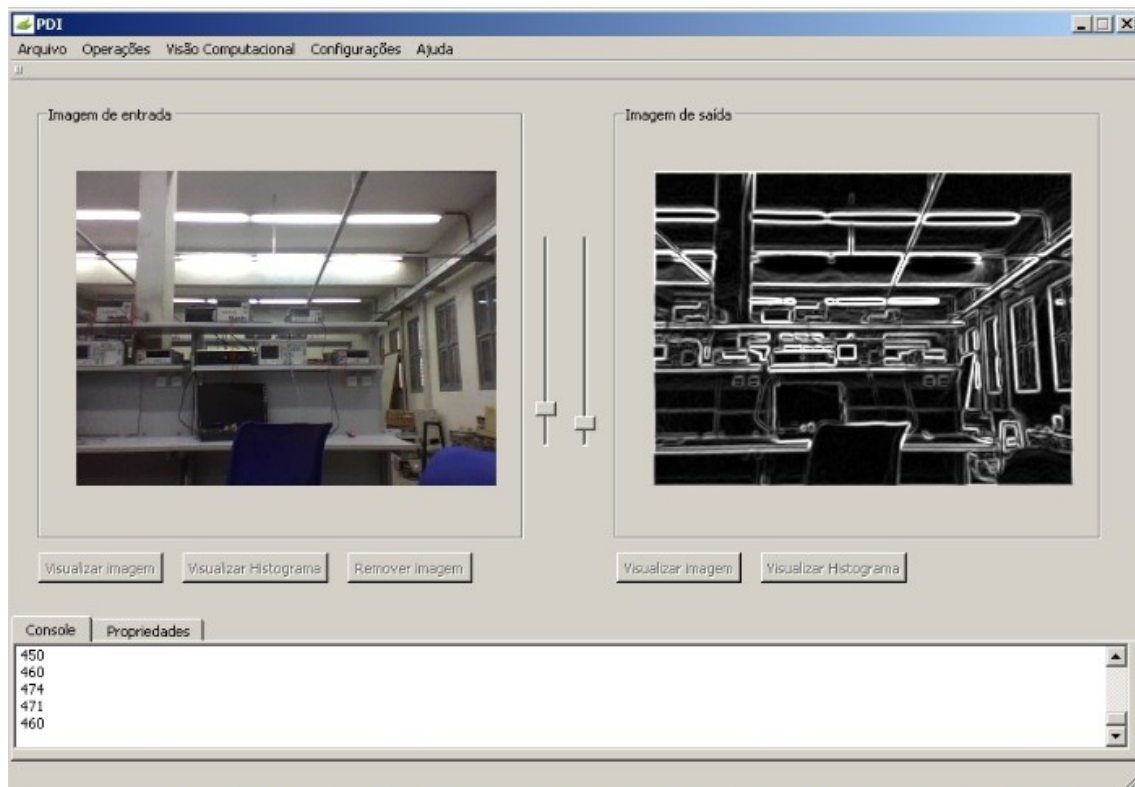
- Remoção de ruído, através do filtro Gaussiano, com máscara $\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$;
- Realce de bordas, através do gradiente, com a máscara do filtro de Sobel $\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$ e $\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$;
- Non-maximum supression, conectando as bordas com as bordas dos pixels vizinhos, através da fase;
- Duplo Thresholding, removendo as bordas derivadas de ruído, passando um segunda thresholding mais “relaxado” para o caso de bordas vizinhas;
- Rastreamento, listar as posições das bordas.

Resultados dos passos:

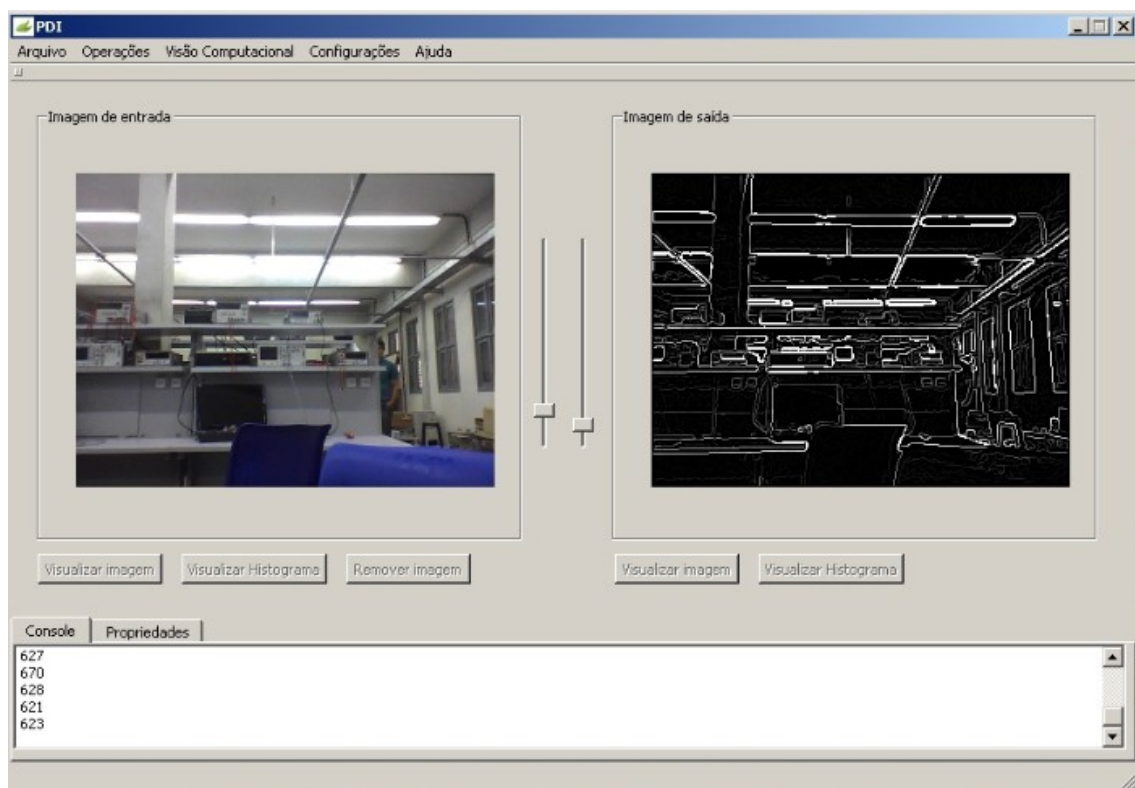
1. Suavização



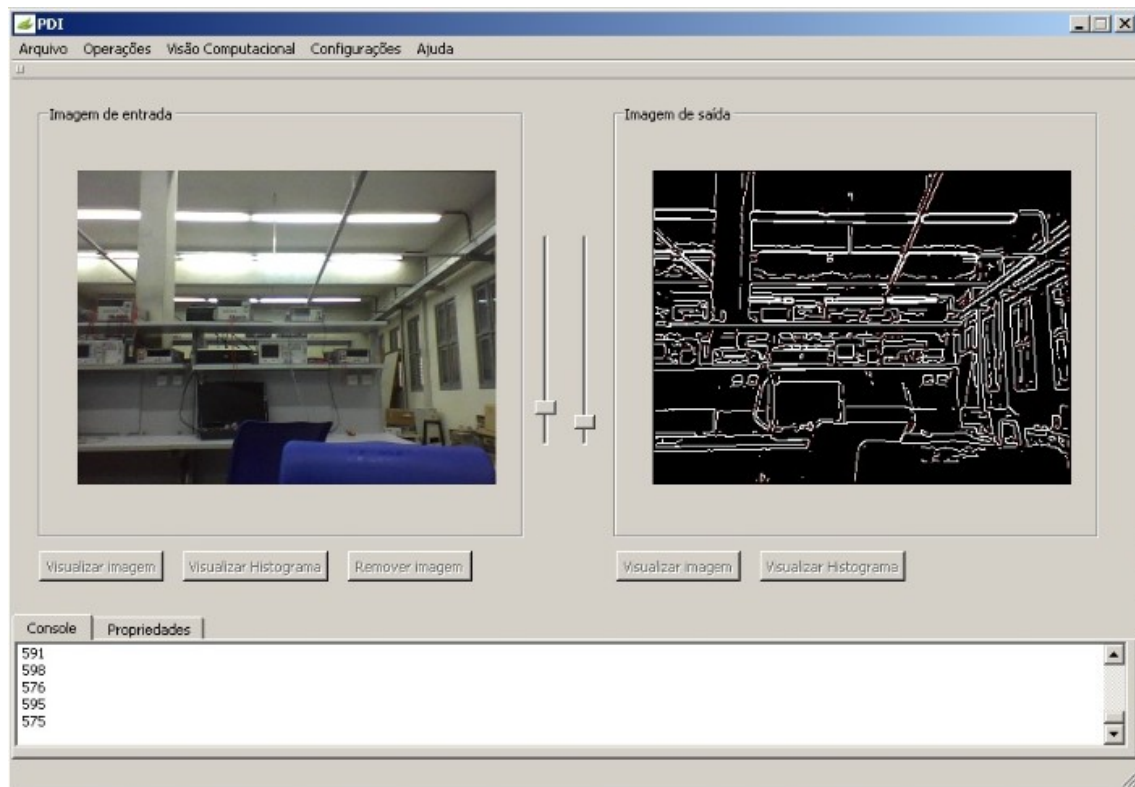
2. Encontrando gradientes



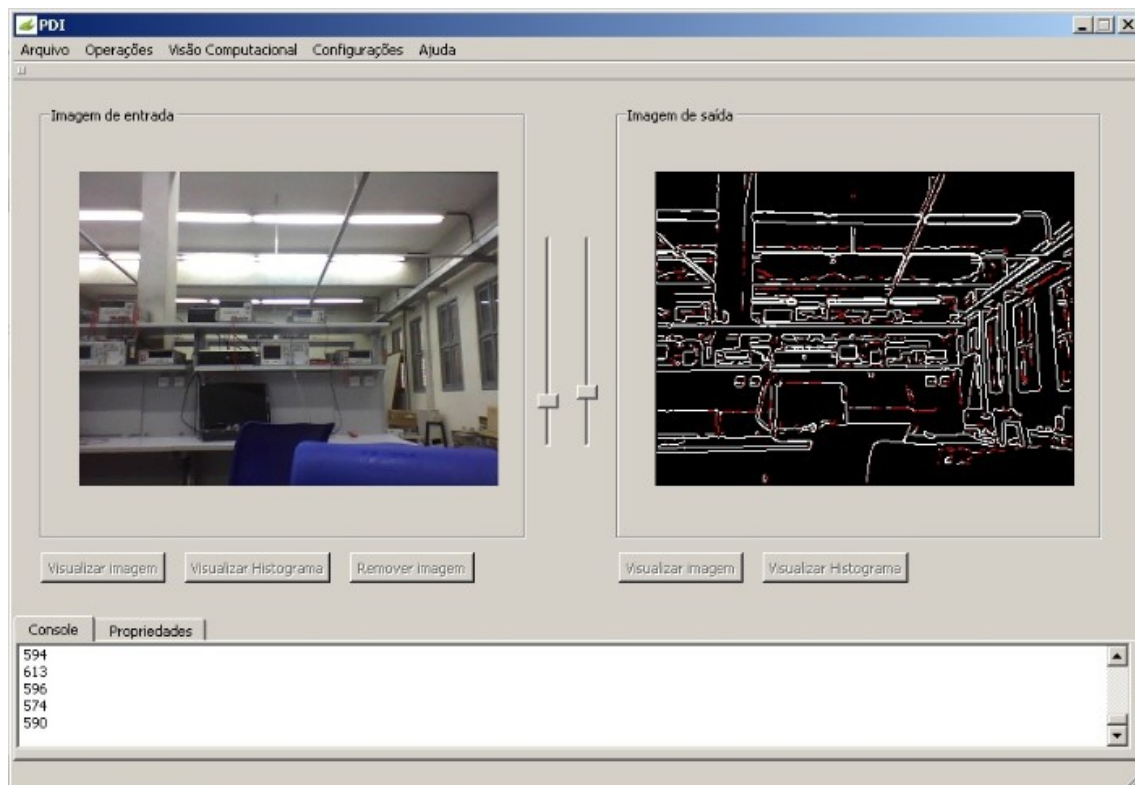
3. Non-maximum suppression



4. Duplo Thresholding



Variação do Thresholding



Método de Frei e Chen

Os seguintes filtros são construídos:

$$\begin{aligned} G_1 &= \frac{1}{2\sqrt{2}} \begin{bmatrix} 1 & \sqrt{2} & 1 \\ 0 & 0 & 0 \\ -1 & -\sqrt{2} & -1 \end{bmatrix} & G_2 &= \frac{1}{2\sqrt{2}} \begin{bmatrix} 1 & 0 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 1 & 0 & -1 \end{bmatrix} & G_3 &= \frac{1}{2\sqrt{2}} \begin{bmatrix} 0 & -1 & \sqrt{2} \\ 1 & 0 & -1 \\ -\sqrt{2} & 1 & 0 \end{bmatrix} \\ G_4 &= \frac{1}{2\sqrt{2}} \begin{bmatrix} \sqrt{2} & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & -\sqrt{2} \end{bmatrix} & G_5 &= \frac{1}{2} \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} & G_6 &= \frac{1}{2} \begin{bmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix} \\ G_7 &= \frac{1}{6} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} & G_8 &= \frac{1}{6} \begin{bmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{bmatrix} & G_9 &= \frac{1}{3} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \end{aligned}$$

Onde para cada pixel temos:

$$|\Delta I| = \sqrt{(G_x * I)^2 + (G_y * I)^2}$$

E é encontrado o ângulo e :

$$\cos e = \sqrt{\frac{M}{S}} \quad \text{where } M = \sum_{k \in \{e\}} (G_k * I)^2 \quad \text{and } S = \sum_{k=1}^9 (G_k * I)^2$$

E depois normalizado e utilizado um threshold para o ângulo 45° .

Imagem de entrada:

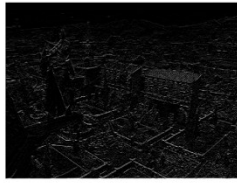


Resultado:

filtro k = 0



filtro k = 1



filtro k = 2



filtro k = 3



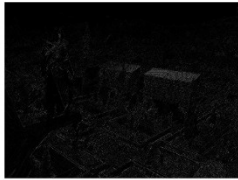
filtro k = 4



filtro k = 5



filtro k = 6



filtro k = 7



filtro k = 8



Referências

- [1] – John Canny. A Computational Approach to Edge Detection. IEEE. 1986.
- [2] – Digital Image Processing. Rafael C. Gonzalez e Richard E. Woods.