

COMPARAÇÃO DE ALGORITMOS DE DETECÇÃO DE BORDAS

MANOEL VIEIRA COELHO NETO*

*SQS 203 Bloco J
Brasília, DF, Brasil

Email: vieiranetoc@gmail.com

1 Objetivos

Este relatório tem como objetivo apresentar os resultados para comparação de diferentes algoritmos de detecção de bordas em relação com um *ground truth* de referência. Assim, aplicamos o método de Sobel, Laplace e Canny para detecção de bordas e comparamos a porcentagem de igualdade entre a saída do algoritmo e o *ground truth*.

2 Introdução

Três algoritmos foram usados para detecção de bordas: Sobel, Laplace e Canny.

Para o operador de Sobel, calculamos as derivadas horizontais e verticais dos pixels da imagem e então tiramos o gradiente G dado por $G = |G_x| + |G_y|$ em que:

$$G_x = \begin{vmatrix} 1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & -0 & +1 \end{vmatrix} * I.$$
$$G_y = \begin{vmatrix} 1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{vmatrix} * I.$$

Onde I é a imagem a ser operada e $G_x G_y$ são matrizes de convolução

De fato o operador de Sobel é um operador de diferenciação discreta, simplesmente computa a aproximação do gradiente de uma imagem, combinando borramento gaussiano e diferenciação.

O operador de Laplace se refere à segunda derivada dos pixels de uma imagem e é um operador de segunda ordem, assim nos pontos de máximos do operador de primeira ordem, a segunda derivada é igual a zero. Então o que deve ocorrer é:

$$G' = 0$$

Ou usamos o laplaciano do pixel, dado pela sua derivada em x e y .

$$\text{Laplace}(f) = \frac{\delta^2 f}{\delta x^2} + \frac{\delta^2 f}{\delta y^2}$$

Mas ambos os anteriores apesar de fazer um bom trabalho não tem retorno satisfatório em termos de porcentagem de bordas achadas, porque ambos dependem de achar a derivada máxima de um pixel e há um erro associado ao cálculo pois, às vezes acaba-se por achar um máximo local ao invés de um global. Então Canny propôs seu método objetivando três critérios principais:

- Baixa taxa de erro: Boa detecção somente em bordas existentes
- Boa localização: a distância entre pixels de bordas detectados e os reais tem de ser minimizadas.
- Resposta Mínima: Somente uma resposta de detector por borda.

O primeiro passo é usar um filtro gaussiano para reduzir o ruído da imagem

Achar o gradiente da imagem, usando um método análogo ao de Sobel, após calcular as matrizes de convolução, após isso, tira-se a magnitude de G , e sua direção θ , onde:
$$\theta = \arctan\left(\frac{G_y}{G_x}\right)$$

E a direção é dada em 4 possíveis ângulos: 0° , 45° , 90° ou 135° .

Aplica-se a supressão de não-máximos, removendo todos os pixels que não são considerados parte de uma borda, assim, somente as linhas finas, que são nesse caso as linhas de interesse - borda - restarão na imagem.

E o passo final é chamado de histerese, onde são aplicados dois *thresholds*, o primeiro verifica se o pixel está acima do valor de *threshold*, se sim, é aceito como uma borda, se não é rejeitado e o segundo verifica se os pixels que estão entre os dois valores estão conectados a uma borda, se sim são bordas, se não, são rejeitados.

3 Metodologia

Os algoritmos de Sobel, Operador de Laplace e Canny usados foram os que estão implementados no site de tutoriais do OpenCV e podem ser verificados nas urls abaixo.

http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/sobel_derivatives/sobel_derivatives.html

http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.html

http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/laplace_operator/laplace_operator.html

E aplicando-se cada um dos filtros a imagens de exemplo e por fim comparando pixel a pixel

a saída com sua *ground truth*, fazemos uma porcentagem de pixels correspondentes entre as duas, para que se possa analisar qual o melhor método de detecção de bordas.

4 Resultados

Para a imagem de entrada:



Temos as seguintes saídas:



Sobel



Laplace



Canny

Nossa imagem de *ground truth* que nos dá as bordas "reais" da imagem é:



Fazendo a comparação 1-1 de cada uma das saídas com o *ground truth* para todas as imagens de exemplo temos a seguinte tabela de porcentagem de correspondência:

Tabela 1: Precisão dos algoritmos

Precisão	46	140	208	212	217	221
Sobel	0.2	0.006	0.01	0.006	0.01	0.10
Laplace	0.3	0.07	0.05	0.11	0.09	0.20
Canny	0.9	0.75	0.76	0.76	0.72	0.88

5 Conclusão e Discussões

Assim foi constatado a maior precisão do operador de Canny. De fato há mais cuidado sobre os resultados do gradiente e a suavização de ruídos que possam atrapalhar. Para imagens com uma variação muito alta de cor entre duas regiões é possível perceber que até mesmo Canny tem um certo nível de falha, uma vez que todos os operadores aplicados no presente processo dependem do gradiente, que nada mais é que a derivada do pixel em x e y. Então, caso haja uma variação brusca, o filtro tende a ser impreciso naquela região de variação.

Referências

OpenCV Docs (n.d.).

<http://docs.opencv.org/2.4/>

(*OpenCV Docs*, n.d.)

[http://docs.opencv.org/doc/tutorials/
imgproc/imgtrans/sobel_derivatives/sobel_
derivatives.html](http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/sobel_derivatives/sobel_derivatives.html)

[http://docs.opencv.org/doc/tutorials/
imgproc/imgtrans/canny_detector/canny_
detector.html](http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.html)

[http://docs.opencv.org/doc/tutorials/
imgproc/imgtrans/laplace_operator/
laplace_operator.html](http://docs.opencv.org/doc/tutorials/imgproc/imgtrans/laplace_operator/laplace_operator.html)