PROJETO DEMONSTRATIVO 4 - DETECÇÃO DE BORDAS

Samuel Venzi Lima Monteiro de Oliveira
14/0162241*

**SQN 208*
Brasília
Brasil

Email: samuel.venzi@me.com

1 Objetivos

O objetivo deste projeto é entender e utilizar três técnicas de detecção de bordas (Canny, Sobel e Laplaciano), e após a detecção das bordas a partir de imagens pré-determinadas comparar com o resultado modelo de cada uma para verificar a precisão de cada método.

2 Introdução

Detecção de bordas tem uma vasta aplicação prática no mundo e é um dos precursores para as técnicas de detecção de objetos e segmentação de imagens. A aplicação vai desde a área médica até a filtros simples de editores de imagens.

No geral o processo de detecção se baseia no cálculo de taxas de variação de cor pela imagem, ou seja, uso de de derivadas. Essa técnica realça as bordas na direção do cálculo da derivada. Com os valores dessas derivadas em cada um dos pontos da imagem pode-se manipular a imagem para extrair as bordas.

Apesar de ser relativamente simples o entendimento da detecção de bordas, as condições da imagem podem atrapalhar substancialmente a técnica e produzir resultados indesejáveis. Uma imagem com muito ruído produzirá variações que poderão ser entendidas erroneamente como bordas. Então, antes da realização do processo de detecção de bordas propriamente dito, é aplicado, na maioria da vezes, um filtro de suavização gaussiano para a redução de ruído.

A técnica de Canny é bastante utilizada devido a imagem que é retornada pois ela tem baixa taxa de erro, as bordas são mais próximos das bordas reais e um único ponto de borda é retornado.

Sobel e Laplaciano por outro lado retornam uma imagem com bordas mais largas e e inclusive detectam texturas na imagem, na maioria das vezes ignorado por Canny, dependendo do parâmetro de *threshold*.

O OpenCV facilita o trabalho por ter as três técnicas já implementadas, além a suavização gaussiana.

3 Materiais e Metodologia

3.1 Materiais

- Computador com ambiente Linux (Ubuntu)
- Imagens para a detecção de bordas.
- OpenCV

3.2 Metodologia

Para aplicar as técnicas às imagens disponibilizadas, primeiro foi feita a suavização para diminuir o ruído e gerar um resultado mais confiável e a partir de funções já implementadas do OpenCV foram aplicadas às imagens as técnicas de Canny, Sobel e Laplaciano.

Após esse passo, foi desenvolvido um algoritmo que compara os pixels da imagem gerada com os pixels da imagem de referência e feita uma razão para achar a precisão para cada método. $Precisao = \frac{pixels_iguais}{pixels_iguais + pixels_diferentes}$

O resultado acima multiplicado por 100 resulta na porcentagem de precisão.

4 Resultados

Por serem 24 imagens mostrando resultados elas estão em anexo na pasta Resultados no arquivo comprimido de envio.

A tabela a seguir relaciona a precisão calculada para cada imagem e para cada método.

Table 1: precisão (porcentagem)

método	46	140	208	212	217	221
Canny	65.6	44.7	43.0	43.0	41.8	73.3
Sobel	22.8	02.9	03.3	04.7	07.51	16.3
Laplacian	22.4	04.6	04.4	07.1	07.7	17.1

5 Discussão e Conclusões

Nas imagens dos resultados em anexo é possível observar o esperado pela teoria estudada. Para todas as imagens os resultados foram compatíveis e nelas é possível observar a precisão já citada do método de Canny e as bordas mais largas dos métodos de Sobel e Laplaciano.

Com relação à precisão dos métodos, pela Tabela 1, é possível verificar sem dúvida nenhuma que o método mais preciso é o método de Canny, enquanto Sobel e Laplaciano apresentam valores muito parecidos. Resultado esperado, já que Canny retorna um único valor de borda, o que gera uma borda fina e próxima da real coincidente com a imagem referência.

References

Sobel derivatives (n.d.).
`http://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/sobel_derivatives/sobel_derivatives.html`

Canny Edge Detection (n.d.).
`http://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/canny_detector/canny_detector.html`

Laplace Operator (n.d.).
`http://docs.opencv.org/2.4/doc/tutorials/imgproc/imgtrans/laplace_operator/laplace_operator.html`

Edge Detection (n.d.).
`https://en.wikipedia.org/wiki/Edge_detection`