DS878-N Processamento Digital de Imagens

Julio Cesar Mariotto Junior

Algoritmos de detecção de contorno Gaussian, Sobel e Canny

Curitiba

2021

Gaussian

O algoritmo gaussiano, ou filtro gaussiano, é utilizado no pré-processamento da imagem, para a suavização de distorções e ruídos, com a diferença de não preservar as arestas uma vez que não considera a diferença das intensidades. O quanto a imagem será suavizada está relacionado ao desvio padrão sigma, isto é, quanto maior o sigma, mais a imagem é suavizada, não dependendo muito do parâmetro referente a dimensão da janela. Quanto maior o sigma, maior o número de pixels cujo valor é diferente de zero, o que leva os pixels vizinhos a terem maior influência em cada ponto, realizando uma suavização maior na imagem.

Sobel

O filtro Sobel é uma operação utilizada no processamento de imagem, aplicada sobretudo em algoritmos de detecção de contornos. Consiste num operador que calcula diferenças finitas, dando uma aproximação do gradiente da intensidade dos pixels da imagem. Em cada ponto da imagem, o resultado da aplicação do filtro Sobel devolve o gradiente ou a norma deste vector. Como as variações claro-escuro intensas correspondem a fronteiras bem definidas entre objetos, consegue-se fazer a detecção de contornos.

Canny

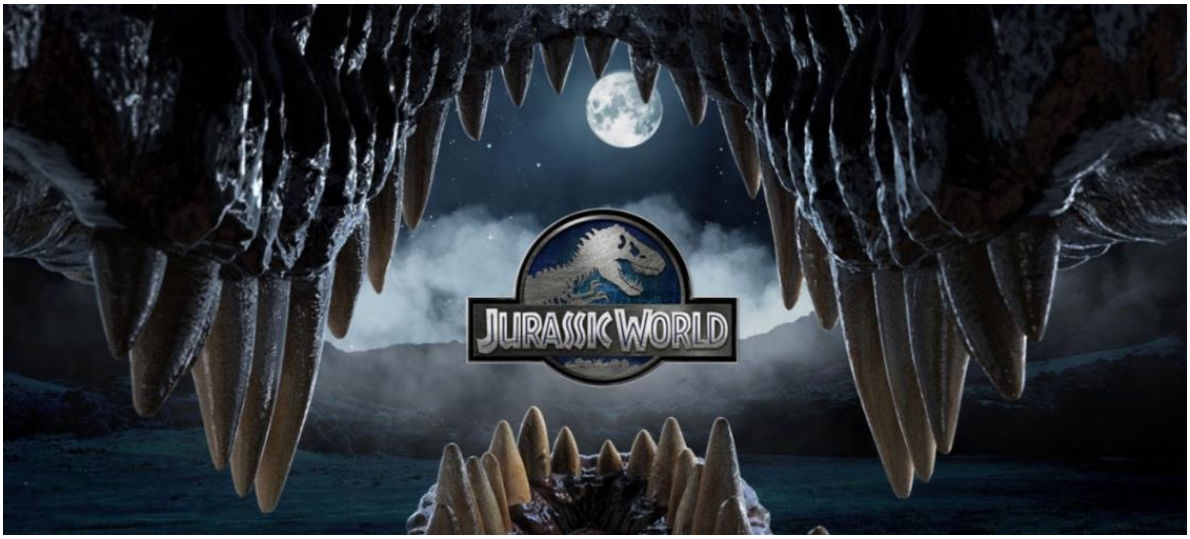
O detector de bordas de Canny utiliza um algoritmo multe estágios para detectar uma ampla margem de bordas na imagem. John Canny propôs que o detector de bordas ótimo deveria respeitar os seguintes parâmetros:

Boa Detecção - O algoritmo deve ser capaz de identificar todas as bordas possíveis na imagem.
Boa Localização - As bordas encontradas devem estar o mais próximo possível das bordas da imagem original.
Resposta Mínima - Cada borda da imagem deve ser marcada apenas uma vez. O ruído da imagem não deve criar falsas bordas.

A função ideal para o detector de Canny é descrito pela soma de quatro termos de exponenciais, que pode ser aproximada pela primeira derivada de uma gaussiana.

Teste Prático

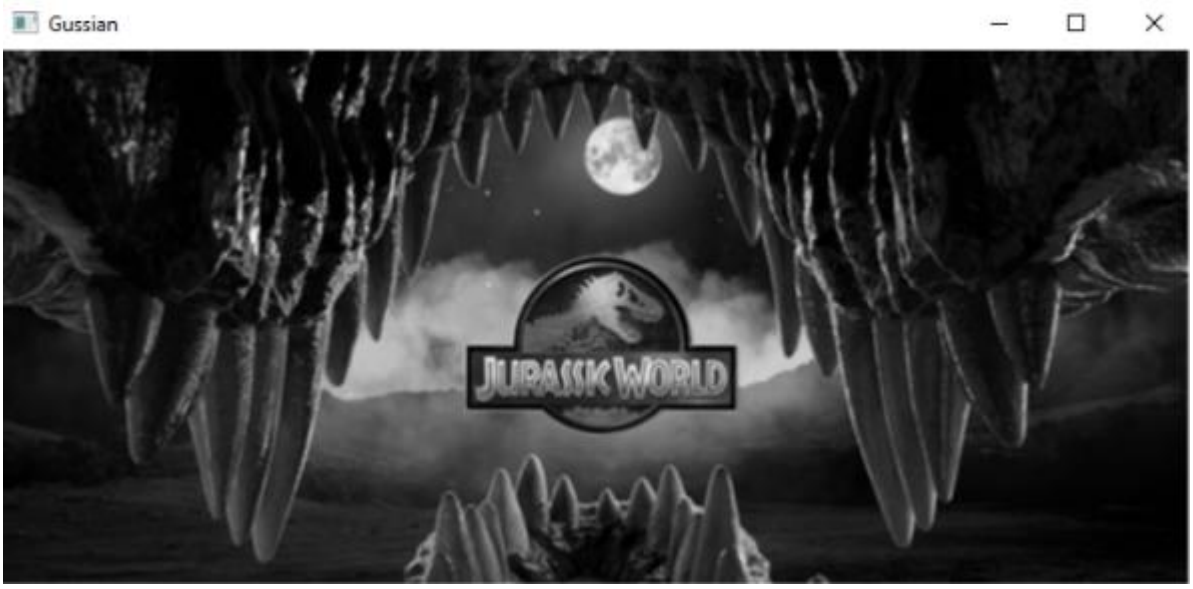
Imagem selecionada para o experimento:



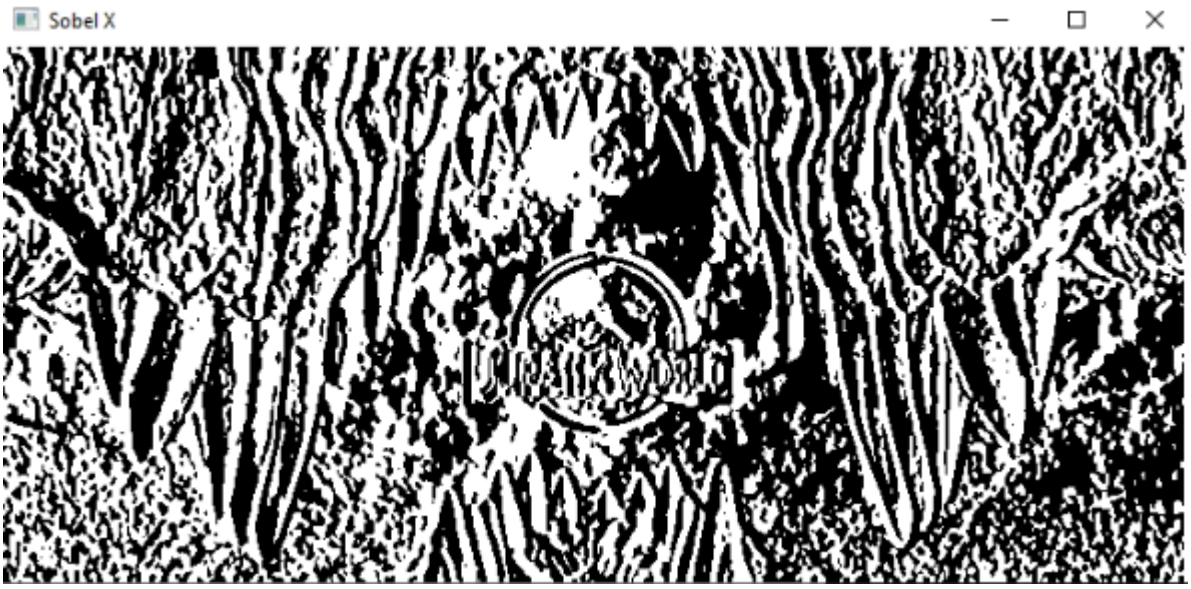
Desenvolvi uma aplicação no Python com o OpenCV que utiliza os 3 algoritmos para realizar o procedimento de detecção das bordas e mostra o passo a passo até a imagem final. Aqui está o código em Python:

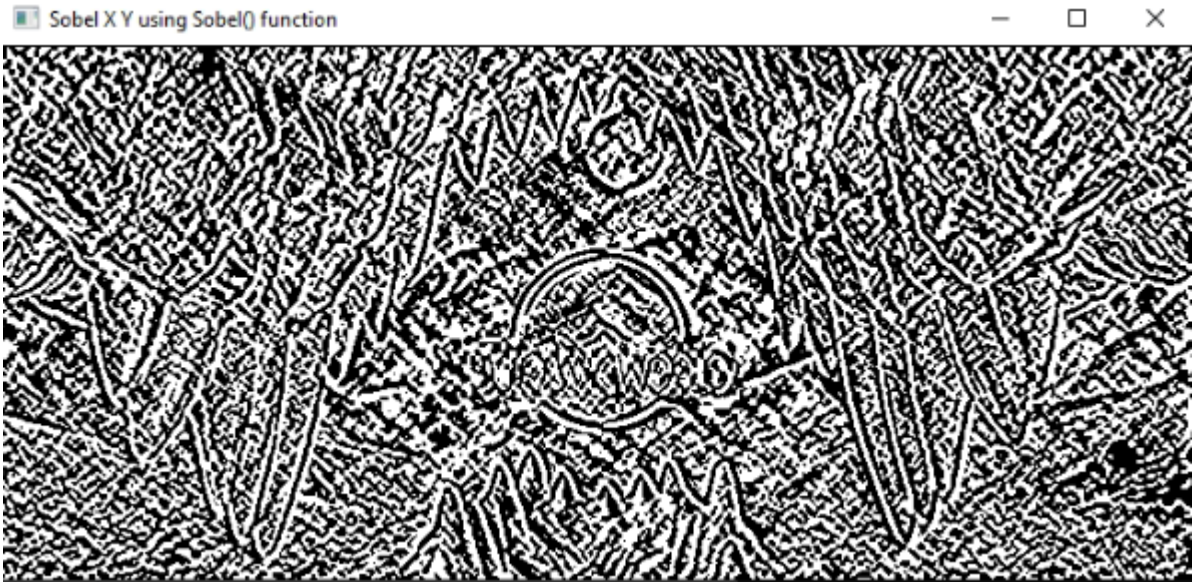
```python
import cv2

img = cv2.imread('jurassic.bmp')

cv2.imshow('Original', img)
cv2.waitKey(0)

img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
img_blur = cv2.GaussianBlur(img_gray, (3,3), 0)

cv2.imshow('Gussian', img_blur)
cv2.waitKey(0)

sobelx = cv2.Sobel(src=img_blur, ddepth=cv2.CV_64F, dx=1, dy=0, ksize=5)
sobely = cv2.Sobel(src=img_blur, ddepth=cv2.CV_64F, dx=0, dy=1, ksize=5)
sobelxy = cv2.Sobel(src=img_blur, ddepth=cv2.CV_64F, dx=1, dy=1, ksize=5)

cv2.imshow('Sobel X', sobelx)
cv2.waitKey(0)
cv2.imshow('Sobel Y', sobely)
cv2.waitKey(0)
cv2.imshow('Sobel X Y using Sobel() function', sobelxy)
cv2.waitKey(0)

edges = cv2.Canny(image=img_blur, threshold1=100, threshold2=200)

cv2.imshow('Canny Edge Detection', edges)
cv2.waitKey(0)
cv2.destroyAllWindows()

```

No pré-processamento, convertemos a imagem para a escala de cinza e aplicamos o algoritmo de Gaussian para suavizar a imagem e prepara-la pra os outros algoritmos.

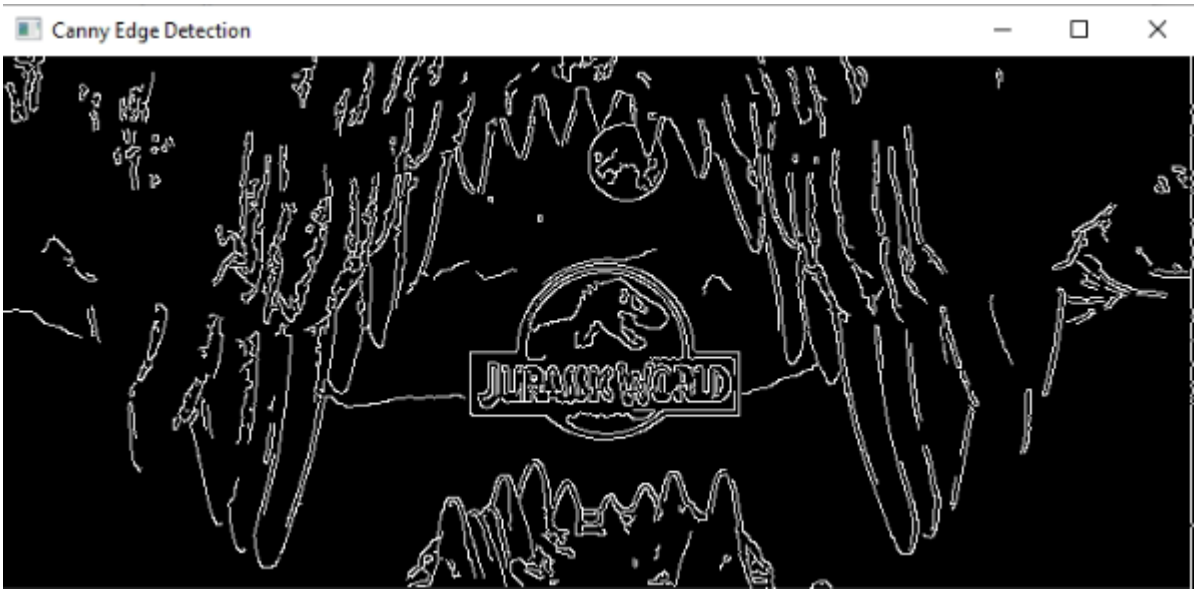


Logo após podemos observar o algoritmo de Sobel realizando a detecção de bordas sobre o eixo X, Y e XY





Por fim temos o algoritmo de Canny sobre a imagem pré-processada de Gaussian.



Conclusão

Após analisar as imagens, percebemos que os algoritmos de Sobel e Canny dependem do filtro de Gaussian para obter os seus resultados. Entre tanto, é notável a qualidade do algoritmo de Canny para destacar as bordas da imagem, já o algoritmo de Sobel tem uma sensibilidade maior nas áreas de sombreamento, talvez seja preciso algum ajuste pra deixar o seu resultado mais preciso.