Canny Edge Detector

Alex Eagles

Contents

Table of Figures

How they work

The Canny Edge Detector method uses the Sobel Edge Detector and adds some enhancements. First of all, in Canny edge we apply Gaussian blur on the image because the reducing the noise can make detecting the edge better, after that we calculate the gradients that we did in the Sobel detector in X and Y directions, we get the magnitude and the angle, after that we do non-maximum suppression and this done by looking at the magnitude and direction of an edge that were got using Sobel and we compare it with its neighbors if the pixel magnitude is not the largest within its neighbors it will be suppressed (set to zero), in the end we do something called double thresholding and in this one we put two thresholding points (high and low one) if a pixel is higher than the high one it will be set to one and if the pixel is lower the low one it will be set to zero but if it was in the middle it will be checked if it is connected to any pixel that its magnitude is higher than the high one and this helps to connect some edges that represent a shape but they were about to be disconnected because of their magnitudes.

What it is Used for and Examples

Canny edge detector is used in image processing to detect edges in many applications, and it can be considered an enhancement to Sobel operator or the next thing to do after Sobel

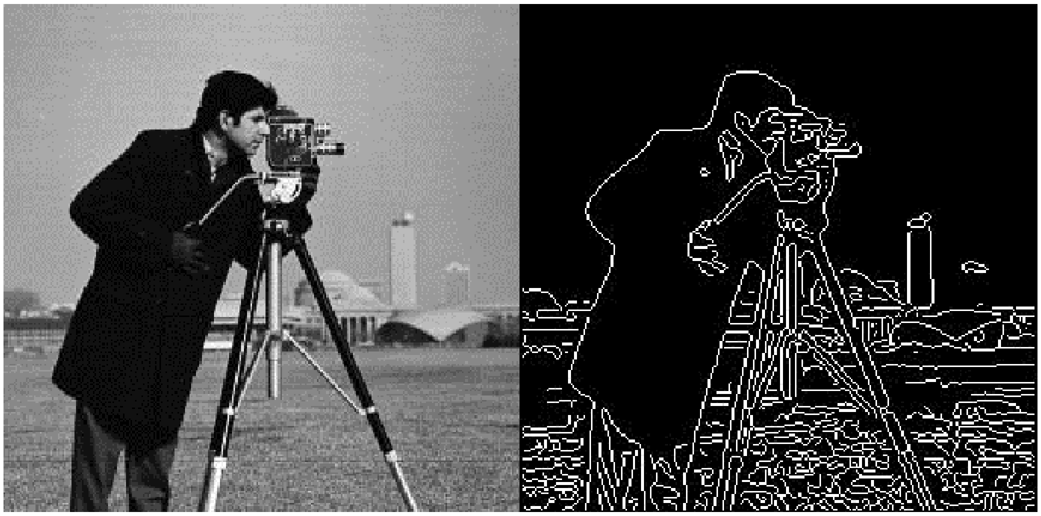
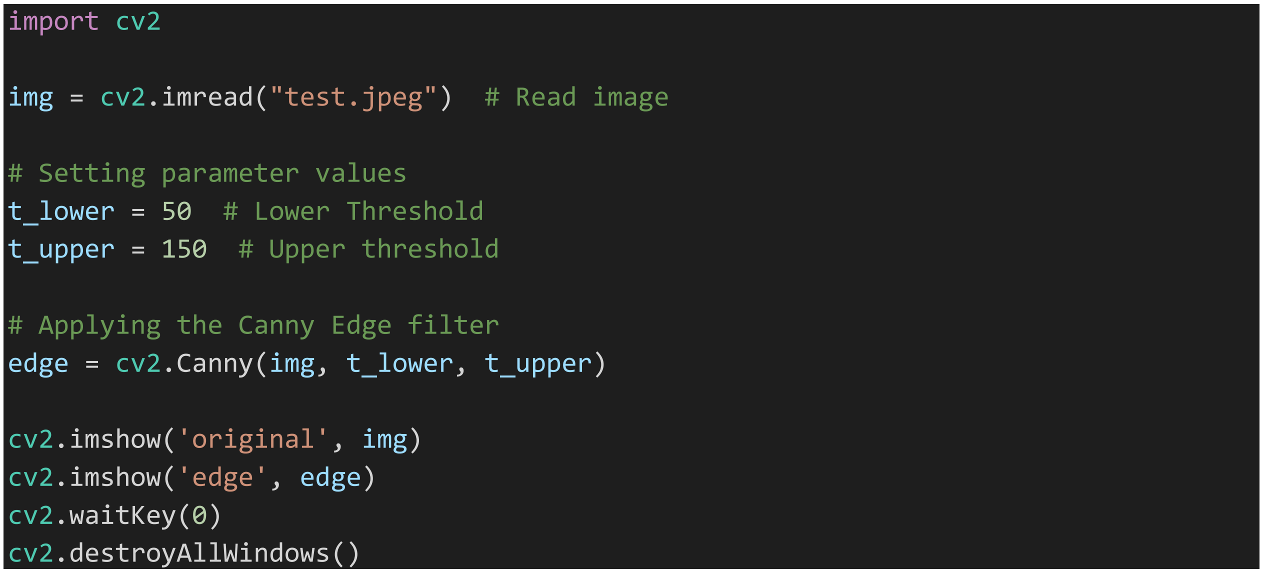


Figure 1 Canny Edge Detector

Python Code



```python
import cv2

img = cv2.imread("test.jpeg")  # Read image

# Setting parameter values
t_lower = 50  # Lower Threshold
t_upper = 150  # Upper threshold

# Applying the Canny Edge filter
edge = cv2.Canny(img, t_lower, t_upper)

cv2.imshow('original', img)
cv2.imshow('edge', edge)
cv2.waitKey(0)
cv2.destroyAllWindows()
```