



ELEC230-202122 ASSIGNMENT 4

Basic Image Processing using OpenCV

Abstract

This report discusses the basic tools of image processing, including Gaussian filter, normalised box filter, Sobel operator, Canny edge detection, Hough line transform, and k-nearest neighbors algorithm for background subtraction, with the help of OpenCV 4.2 and C++.

Declaration

I confirm that I have read and understood the University's definitions of plagiarism and collusion from the Code of Practice on Assessment. I confirm that I have neither committed plagiarism in the completion of this work nor have I colluded with any other party in the preparation and production of this work. The work presented here is my own and in my own words except where I have clearly indicated and acknowledged that I have quoted or used figures from published or unpublished sources (including the web). I understand the consequences of engaging in plagiarism and collusion as described in the Code of Practice on Assessment (Appendix L).

9th May 2022

Contents

1	Background	2
2	Part A Discussion	2
2.1	a	2
2.2	b	2
2.3	c	3
2.4	d	3
2.5	e	4
2.6	f	4
2.7	g	5
3	Part B Discussion	5
	References	6

List of Figures

1	Blurring with different standard deviation	2
2	Sobel operator with normalised box filter (left), and with Gaussian filter (right)	3
3	Sobel operator (left) versus Canny edge detection (right)	4
4	Edges of doorway	5

1 Background

Computer vision is an important tool for mobile robots to perceive the surrounding environment. Unlike humans, however, the images captured by cameras need to be processed before they can be understood by robots, which requires the use of image processing techniques.

This assignment explores the basic tools of image processing with OpenCV 4.2 and C++.

2 Part A Discussion

2.1 a

Kernel is the area that is scanned around a pixel. Larger amount of pixels surround the pixel of interest will be scanned for a larger kernel. If the kernel size increases, the image will become more blurred because the size corresponds to the standard deviation, σ , of Gaussian distribution. In other words, increasing kernel size decreases the weight of the pixels around centre and increases the weight of the distal pixels. This gives a blurrier image.

2.2 b

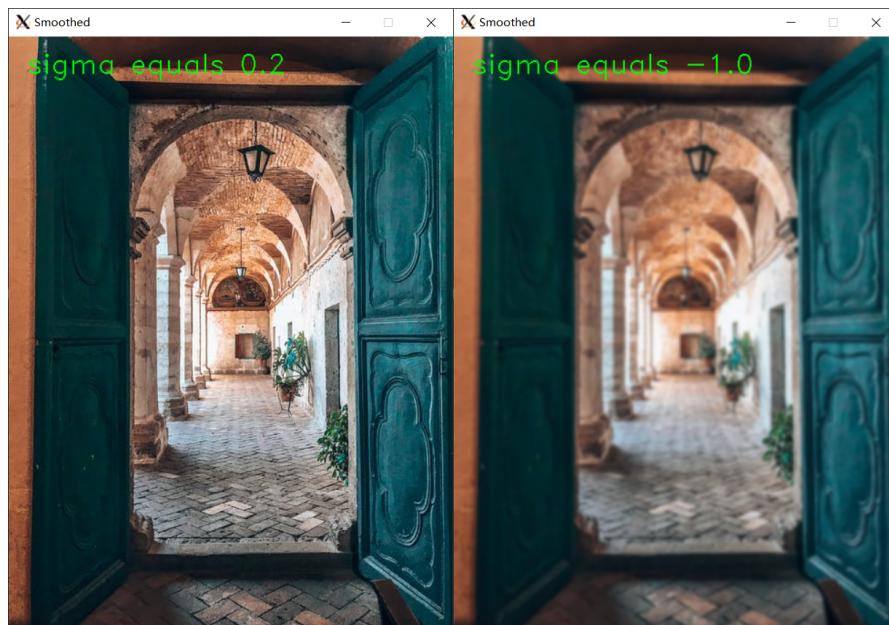


Figure 1: Blurring with different standard deviation

Standard deviation, σ , dictates the width of the Gaussian distribution curve inside the kernel. The standard deviation equal to 0.2 has a more concentrated distribution in the center than -1 . Less centralized distribution gives blurrier image.

Note that the standard deviation cannot be negative. Here -1 is a valid value because it is squared in the formula.

$$G(x, y) = Ae^{-\frac{x^2+y^2}{2\sigma^2}}$$

2.3 c

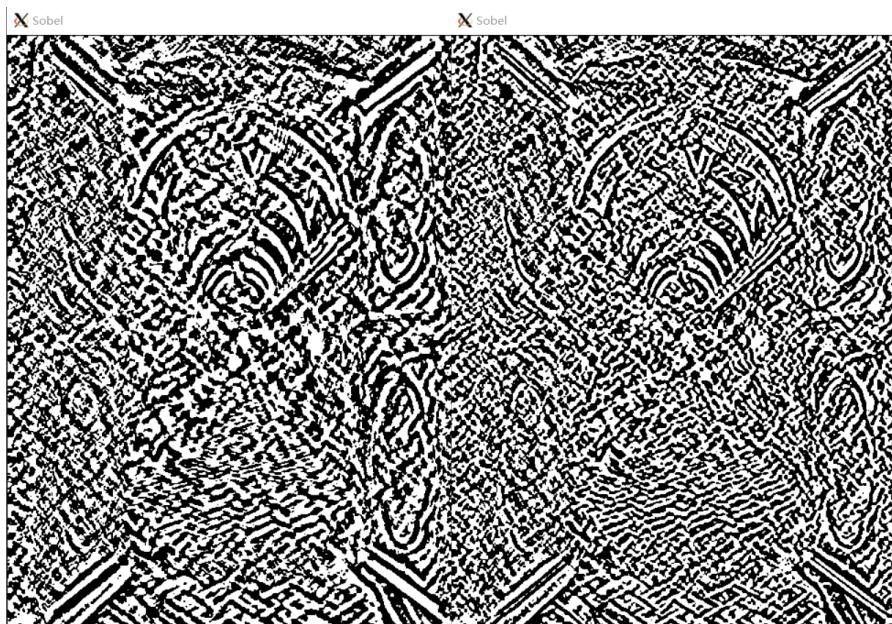


Figure 2: Sobel operator with normalised box filter (left), and with Gaussian filter (right)

Sobel operator with Gaussian smoothing gives more clear edges than with normalised box smoothing. This is because the box smoothing is not a low-pass filter so will cause high frequency artifacts [1]. In other words, the box smoothing cannot produce clear edges. Thus, Gaussian smoothing is better for edge detection.

2.4 d

From observations, increasing the minimum threshold reduces decreases the amount of detected edges. The Canny edge detection algorithm is divided into

four stage, the last stage is Hysteresis Thresholding [2]. Hysteresis Thresholding decides which are really edges based on its intensity gradient calculated in the second stage or its connectivity. Possible edges with intensity gradient below the minimum threshold are discarded.

In addition, The larger the kernel size of Canny, the weaker the effect obtained by adjusting the minimum threshold by the same magnitude. This two parameters hold each other in check, allowing precise control of the amount of edges obtained.

2.5 e

After removing the blurring step which is the first stage of the Canny algorithm, many subtle edges are detected. This reflects the purpose of a smoothing; to remove the small noises and retain the important information. By blurring out the tiny edges in the image, it is more likely that the Canny edge detection yields only the edges of interest.

2.6 f

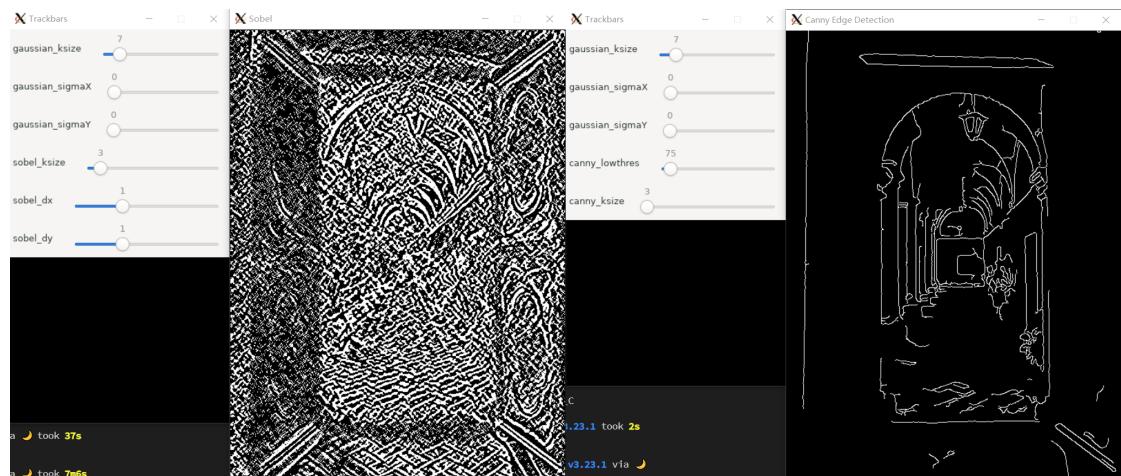


Figure 3: Sobel operator (left) versus Canny edge detection (right)

The Canny edge detection is built on top of the Sobel operator. Edges yields from Sobel operator are thick and contains noises. Moreover, Sobel operator cannot be controlled for getting wanted edges. The Canny edge detection further developed Sobel operator to solve these problems by adding three additional stages which are finding intensity gradient of the image, non-maximum suppression, and hysteresis thresholding [2]. To be more specific, the non-maximum suppression yields thin edges, and the hysteresis thresholding extracts the edges of interest based on its intensity gradient and connectivity.

2.7 g

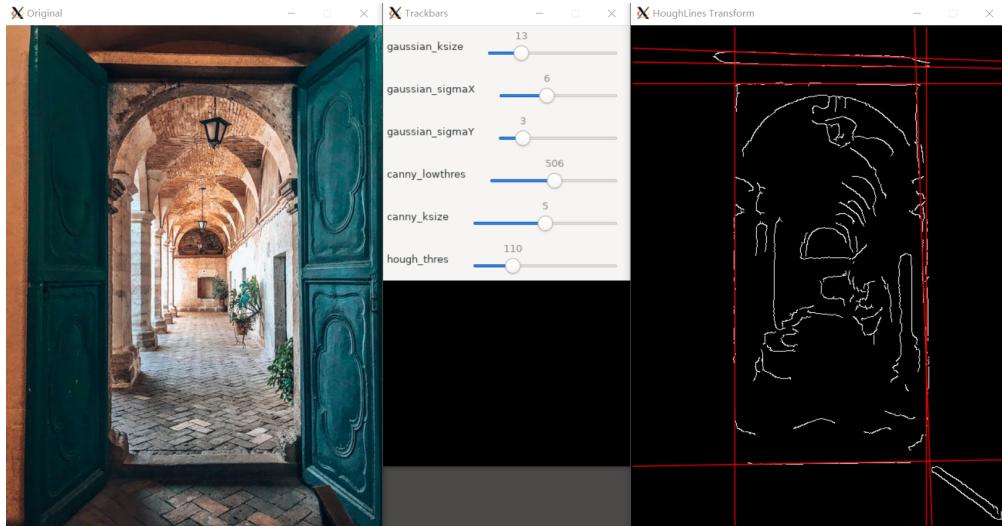


Figure 4: Edges of doorway

The mobile robot can recognise the doorway and pass through it with the help of image processing. The red lines in Fig 4 represent the straight edges of interest, which form the edges of a doorway. There are two redundant red lines at the top but does not matter.

The parameters shown in the middle of Fig 4 are only for this image. The robot camera captures a stream of images split into frames. This means that the parameters should be different for each frame to extract the edges of doorway to ensure that the robot can pass. Dynamic thresholding is a solution for this situation.

3 Part B Discussion

The k-nearest neighbors (KNN) algorithm is a supervised machine learning method. It is a classic method for background subtraction.

It can be noticed that moving objects sometimes appear semi-transparent when using KNN. The explanation for this phenomenon is simple. KNN divides pixels into two categories: foreground and background. The category of a pixel can be represented by the neighbouring pixels. KNN incorrectly classifies some background pixels as foreground pixels, and because of the last sentence, a misclassification can infect the pixels around it, which results in blocks of background pixels appear on the moving objects.

Background subtraction can help robots to achieve target tracking and obstacle avoidance. Usually the target tracked is moving and by eliminating the back-

ground, the robot can better focus on the moving target. Robot obstacle avoidance is divided into two parts, one is avoiding static obstacles and the other is avoiding dynamic obstacles. Background subtraction helps the robot to identify potential dynamic obstacles. Generally speaking, background is usually a noise for robot applications because The primary purpose of giving robots vision is to identify living or moving things, and methods such as Lidar is enough for sensing background information. This type of noise can be eliminated gracefully thanks to the background subtraction algorithms.

References

- [1] R. Collins, *Penn state cse486 lecture 4 smoothing*, Penn State University. [Online]. Available: <https://www.cse.psu.edu/~rtc12/CSE486/lecture04.pdf> (visited on 08/05/2022).
- [2] Canny edge detection, OpenCV. [Online]. Available: https://docs.opencv.org/4.x/da/d22/tutorial_py_canny.html (visited on 08/05/2022).