Real Time Edge Detection

April 11, 2019

Submitted by Group: 6 of Lab Group: B2

Sakib Ahmed (15.01.04.093)
Shamit Ahmed (15.01.04.101)
Sofia Neherin (15.01.04.107)

Course No: CSE 4228

Course Name: Digital Image Processing Lab

Session: Fall 2018

1 Introduction

Edge detection is extensively used in image segmentation to divide an image into areas corresponding to different objects. In a picture, an edge is normally defined as an abrupt change in colour intensity. Human's eyes use a much more complicated method to find edges. This is because we have two eyes (therefore stereoscopic vision and depth perception) as well as our incredible inference skills (we can "see" the grey square above, despite it being obscured by the circle). Despite this, most computer vision systems must do with one (normally grayscale) camera, so change in colour intensity is the next best thing. Edges occur in parts of the image with strong intensity contrast, which often represent object boundaries. Edges characterize object boundaries useful for identification of object in a scene such as an X-Ray image. Edges characterize object boundaries useful for identification of object in a scene such as an X-Ray image. Determining bone edges is important because it can provide surgeons with important information for diagnosis, which in turn enables them to give better treatment decision to their patients. Edge detection is extensively used in image segmentation to divide an image into areas corresponding to different objects. Image segmentation is widely used in many areas including.

2 Dataset

Our project does not require any dataset.

3 Models

3.1 Model 1

This model is our own model where we implemented real time edge detection using different linear algorithms - Roberts,Sobel, Prewitt, Canny.
Here we made a comparison of all the four edge detection algorithm and showed that Canny is the best edge detection algorithm.

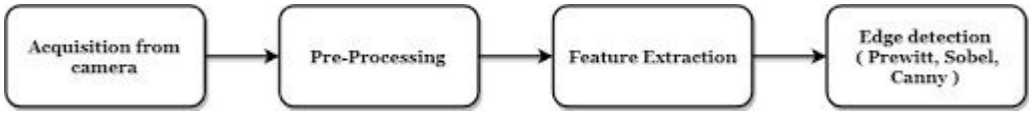

Figure 1: Block Diagram

3.2 Model 2

The morphological gradient is the classical, useful tool in image edge enhancement, but it can be insufficient in real-time applications due its complexity, as are different modifications of this method proposed in literature to obtain clearer edge information. In this model a novel, simple and fast morphological edge detector is presented. It is based on the definition of a non-flat, dynamic structuring element and on the new definition of a simple morphology operation, useful as a generalisation of many image processing operations.
In this model we have implemented a paper[1] where the structuring element SE changes dynamically as it scans through the image.
The steps are defined below:

1. Determining the dynamic SE :
For every pixel (,) m n of the image the values of non-flat SE are computed according to the following
$\{x = m-1, m, m+1; y = n-1, n, n+1 : SE(x,y) = |L(x,y) - L(m,n)|\}$

73	72	76
71	70	89
75	70	73

Table 1: Analyzed image pixel and its neighbourhood covered by SE

3	2	6
1	0	19
5	0	3

Table 2: The values calculated for non-flat SE

2. Applying the function f in the paper the author proposed 3 simple functions:
i. $L(x, y) = max(SE(m, n))$
ii. $L(x, y) = sum(SE(m, n))$
Iii. $L(x, y) = mean(SE(m, n))$
Among the above 3 functions summation function gives the best result.

4 Comparison

Comparing both model 1 which we implemented and model 2 from referred paper we can see that clearly model 2 is better since it gives better edge detection due to its dynamic structuring element. In model 2 we get better edge detection but required time is high since we need to create a structuring element for each individual pixel.

Screenshots of model 1

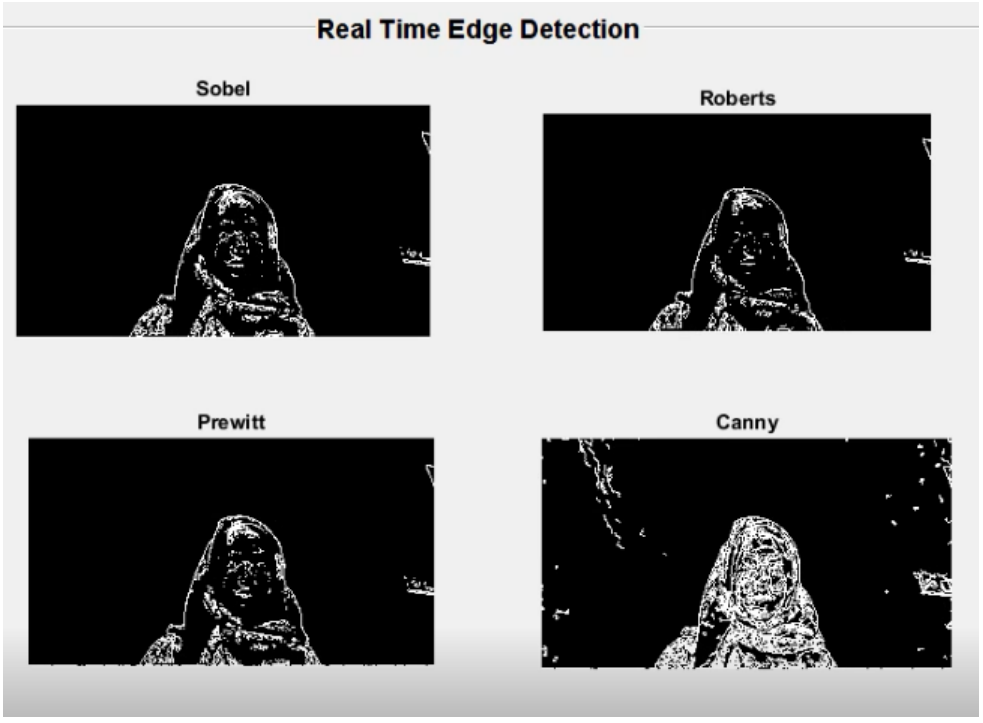

Figure 2: Comparison of Edge Detection Algorithms

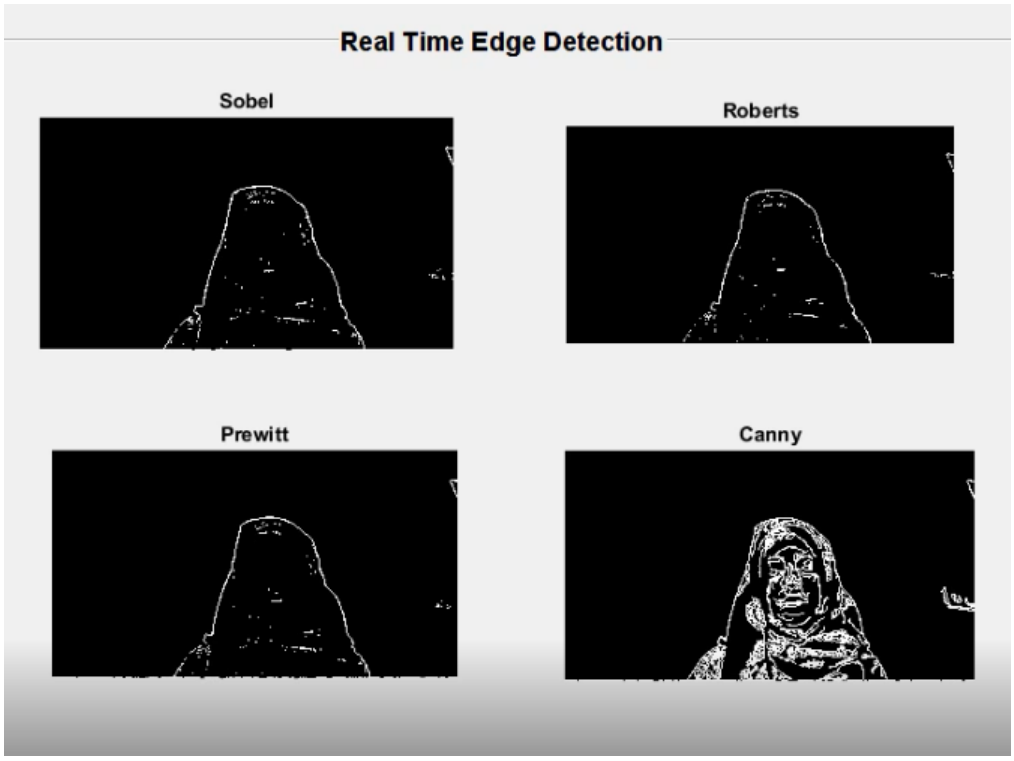

Figure 3: Comparison after applying threshold

Screenshots of model 2

Figure 4: mean function

Figure 5: Sum function

Figure 6: Max function

5 Conclusion

By working on this project we learned that implementing a dynamic structuring element rather than a linear algorithm gives better edge detection while working in real time environment.

References

[1] Kawecki, M. and Putz, B. *Mechatronics 2013*. Springer International Publishing, 2013.