

## Image Segmentation

is one of image analysis methods used to subdivide an image into its regions or objects depending on the type of shapes and objects searched for in the image. Image segmentation is an essential first step in most automatic pictorial pattern recognition and scene analysis tasks.

### Applications of image segmentation

- Inspecting images of electronic boards for missing components or broken connections.
- Detecting faces, facial features and other objects for surveillance.
- Detecting certain cellular objects in biomedical images.

### Segmentation Approaches

Image segmentation algorithms are based on one of two basic properties of gray-level values: **discontinuity** and **similarity**.

- In the first category, the approach is to partition an image based on abrupt discontinuity (i.e. change) in gray level, such as edges in an image.
- In the second category, the approaches are based on partitioning an image into regions that are similar according to a set of predefined criteria.

We shall focus on segmentation algorithms to detect discontinuities such as points, lines and edges. Segmentation methods, studied here, rely on two steps:

1. Choosing appropriate filters that help highlight the required feature(s).
2. Thresholding.

## Point Detection

This is concerned with detecting isolated image points in relation to its neighborhood which is an area of nearly constant gray level.

### 1. Simple method

The simplest point detection method works in two steps:

1. Filter the image with the mask:

-1	-1	-1
-1	8	-1
-1	-1	-1

Then, we take the absolute values of the filtered image.

2. On the filtered image apply an appropriate threshold (e.g. the maximum pixel value).

The next figure shows an example of point detection in a face image using the simple method.

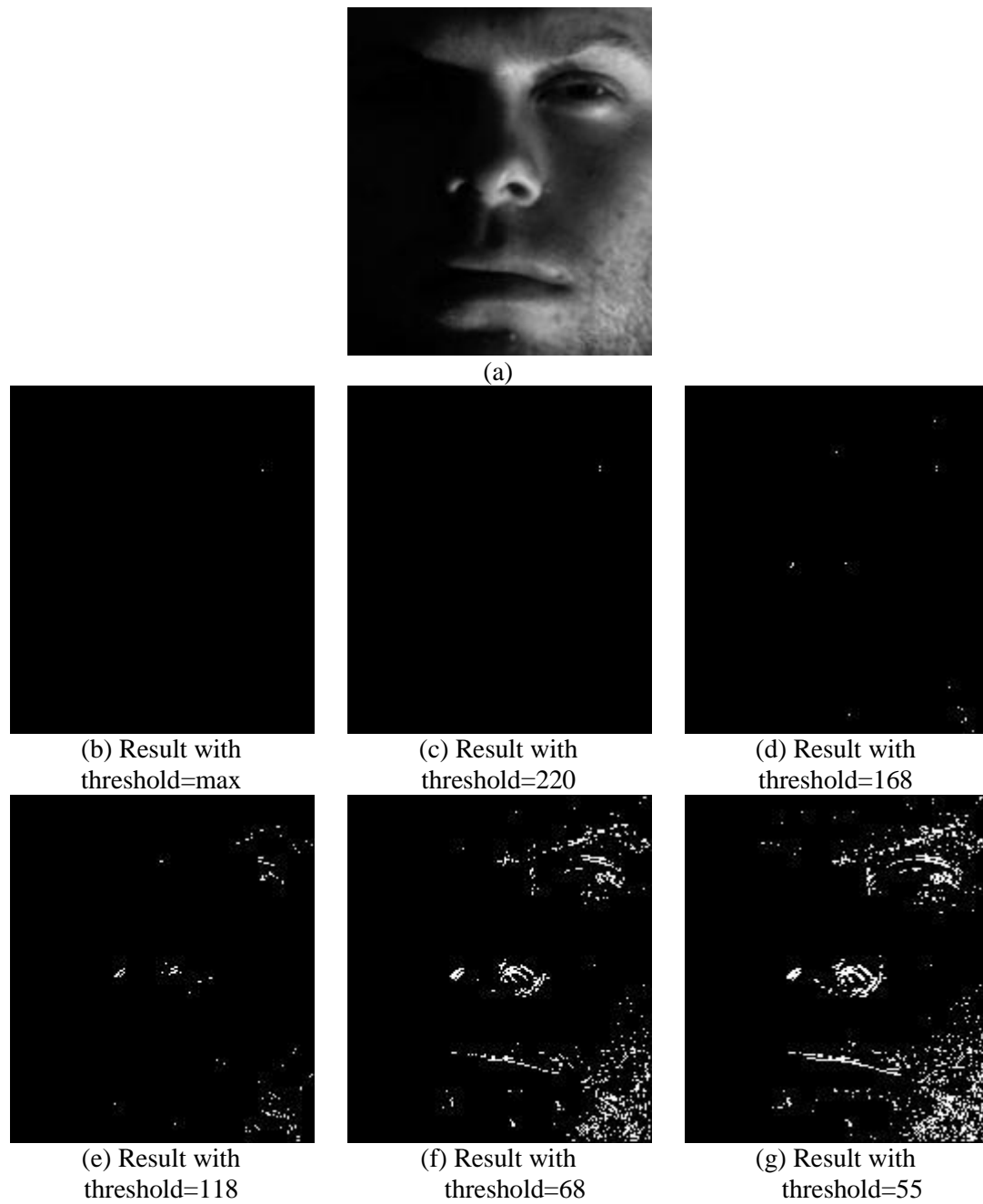


Figure 13.1 Example of point detection using simple method. (a) Original face image.  
(b)-(g) Results with different Thresholds

## 2. Alternative method

An alternative approach to the simple method is to locate the points in a window of a given size where the difference between the max and the min value in the window exceeds a given threshold. This can be done again in two steps:

1. Obtain the difference between the max value (obtained with the order statistics max filter) and the min value (obtained with the order statistics min filter) in the given size mask.
2. On the output image apply an appropriate threshold (e.g. the maximum pixel value).

The figure below shows an example of point detection in a face image using the alternative method.

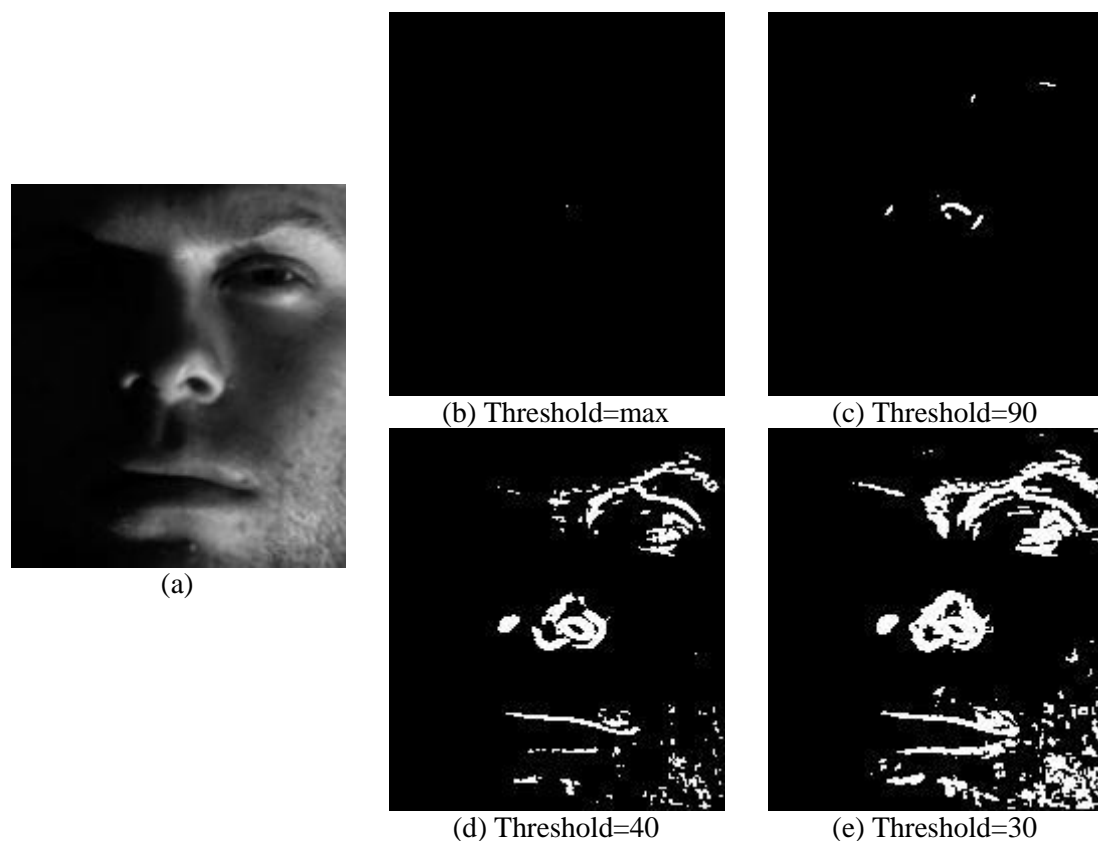


Figure 13.2 Example of point detection using alternative method. (a) Original face image.  
(b)-(e) Results with different Thresholds

## Line Detection

Detecting a line in a certain direction require detecting adjacent points in the image in the given direction. This can be done using filters that yields significant response at points aligned in the given direction.

For example, the following filters

-1	2	-1
-1	2	-1
-1	2	-1

-1	-1	-1
2	2	2
-1	-1	-1

-1	-1	2
-1	2	-1
2	-1	-1

2	-1	-1
-1	2	-1
-1	-1	2

highlight lines in the vertical, horizontal,  $+45^\circ$  direction , and  $-45^\circ$  direction, respectively.

This can be done again in two steps:

1. Filter the image using an appropriate filter.
2. Apply an appropriate threshold (e.g. max value).

The next figure illustrates an example of line detection using the filters above.

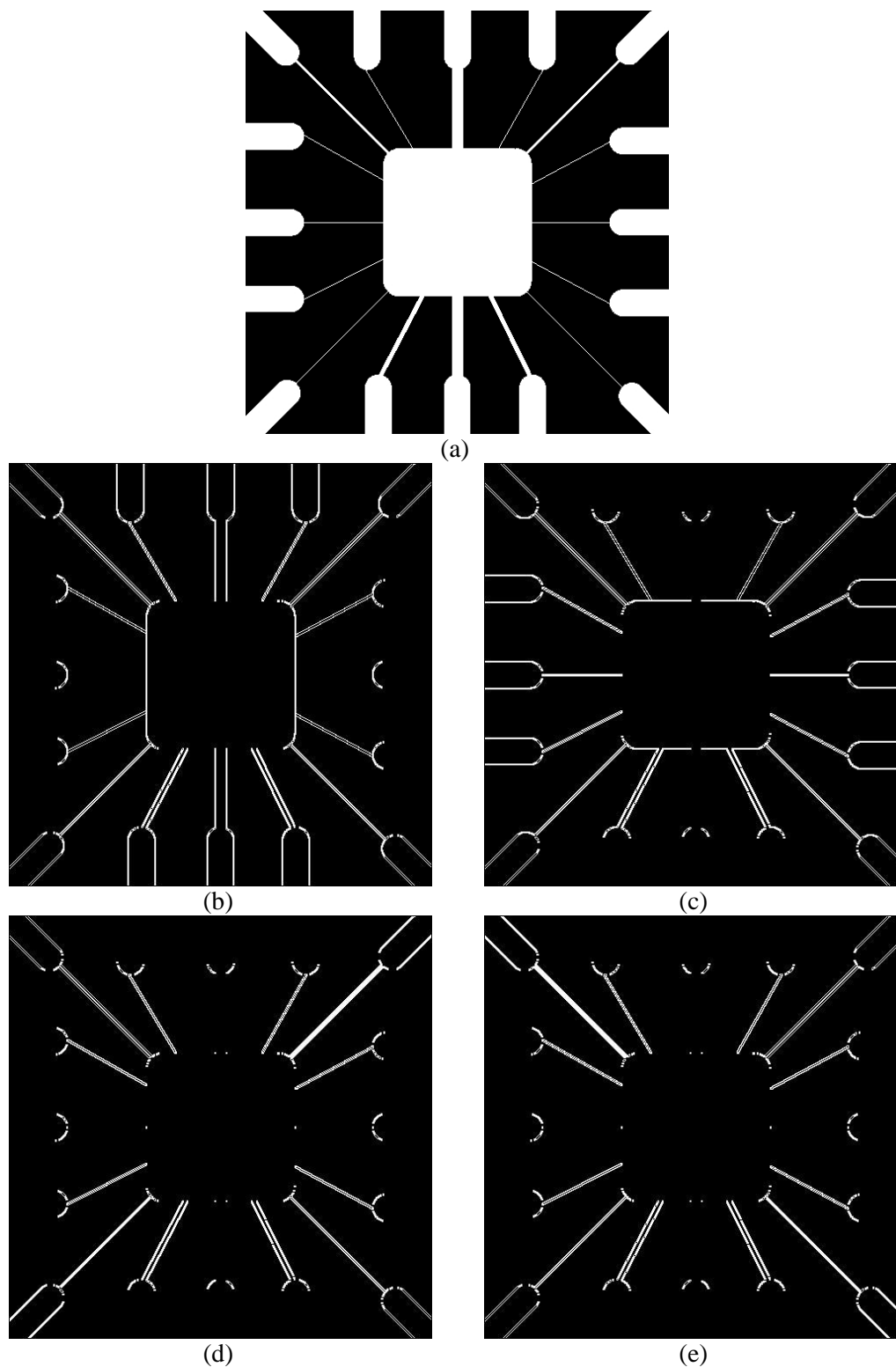


Figure 13.3 Example of line detection. (a) Original image. (b)-(e) Detected lines in the vertical, horizontal,  $+45^\circ$  direction, and  $-45^\circ$  direction, respectively.

## Edge detection

Edge detection in images aims to extract meaningful discontinuity in pixel gray level values. Such discontinuities can be deduced from first and second derivatives as defined in Laplacian filter.

The 1<sup>st</sup>-order derivative of an image  $f(x,y)$  is defined as:

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

Its magnitude is defined as:

$$\nabla f = \sqrt{G_x^2 + G_y^2}$$

Or by using the absolute values

$$\nabla f \approx |G_x| + |G_y|$$

The 2<sup>nd</sup>-order derivative is computed using the Laplacian as follows:

$$\nabla^2 f = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2}$$

However, Laplacian filter is not used for edge detection because, as a second order derivative:

- it is sensitive to noise.
- its magnitude produces double edges.
- it is unable to detect edge direction.

There are 1<sup>st</sup>-order derivative estimators in which we can specify whether the edge detector is sensitive to horizontal or vertical edges or both. We study only two edge detectors namely Sobel and Prewitt edge detectors.

## Sobel edge detector

This detector uses the following masks to approximate the digitally the 1<sup>st</sup>-order derivatives  $G_x$  and  $G_y$ :

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

To detect:

- Horizontal edges, we filter the image  $f$  using the left mask above.
- Vertical edges, we filter the image  $f$  using the right mask above.
- Edges in both directions, we do the following:
  1. Filter the image  $f$  with the left mask to obtain  $G_x$
  2. Filter the image  $f$  again with the right mask to obtain  $G_y$
  3. Compute  $g = \sqrt{G_x^2 + G_y^2}$  or  $g \approx |G_x| + |G_y|$

In all cases, we then take the absolute values of the filtered image, then apply an appropriate threshold.

The next figure shows an example of edge detection using the Sobel detector.



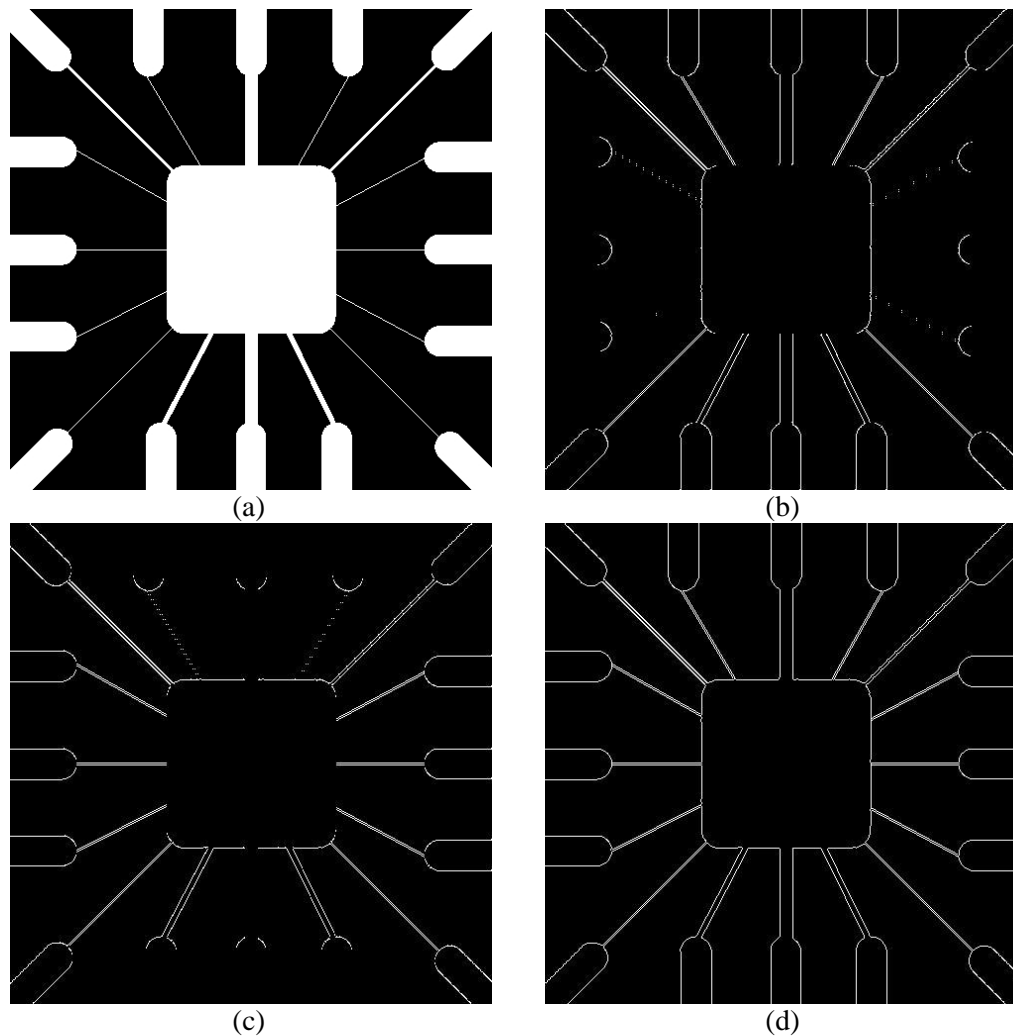


Figure 13.4 Example of Sobel edge detection. (a) Original image. (b)-(d) Edges detected in vertical, horizontal, and both directions, respectively.

### Prewitt edge detector.

This detector uses the following masks to approximate  $G_x$  and  $G_y$ :

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

The steps of applying this detector are the same as that of the Sobel detector.

The next figure shows an example of edge detection using the Prewitt detector.

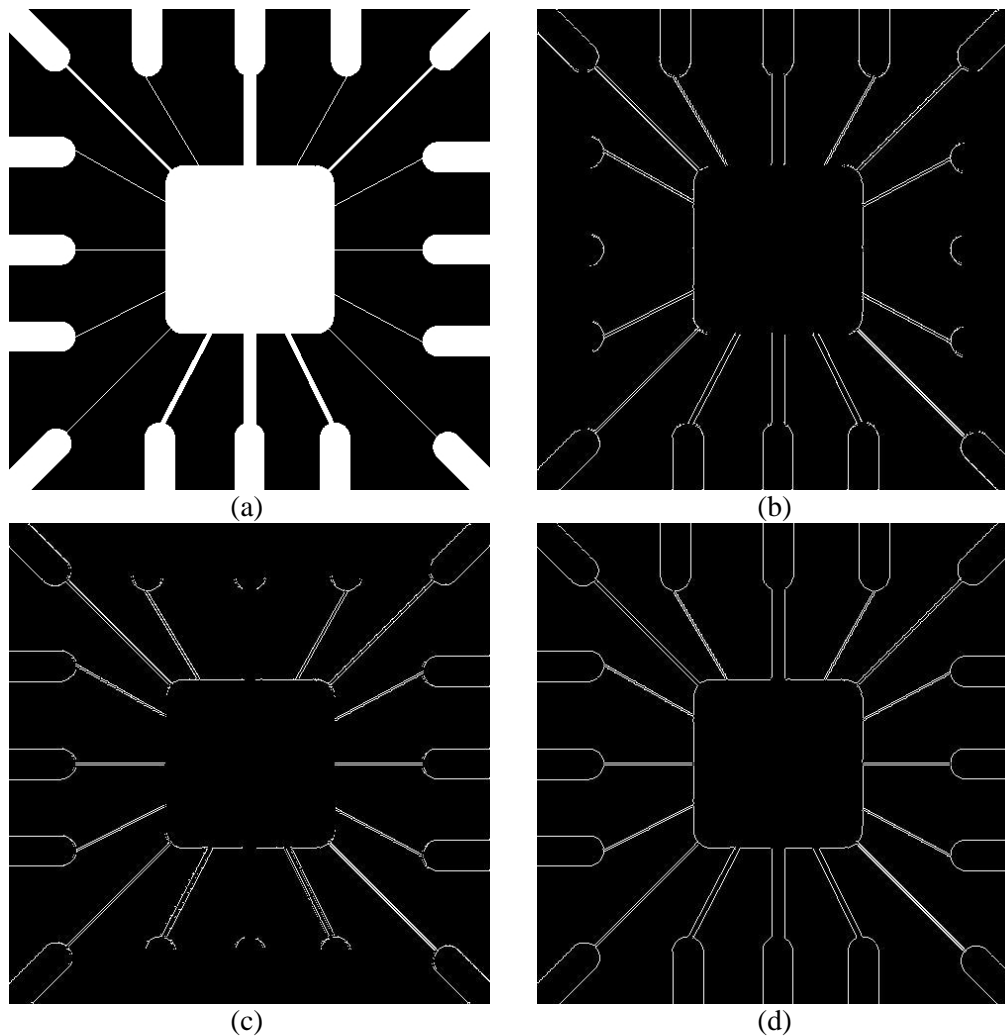


Figure 13.5 Example of Prewitt edge detection. (a) Original image. (b)-(d) Edges detected in vertical, horizontal, and both directions, respectively.

We can see that the Prewitt detector produces noisier results than the Sobel detector. This is because the coefficient with value 2 in the Sobel detector provides smoothing.