CHAPTER 2

Edge Detection

Image Segmentation is the process of partitioning a digital image into multiple regions or sets of pixels [5, 18, 20, 23, 25]. Actually, partitions are different objects in image which have the same texture or color. The result of image segmentation is a set of regions that collectively cover the entire image, or a set of contours extracted from the image. All of the pixels in a region are similar with respect to some characteristic or computed property, such as color, intensity, or texture. Adjacent regions are significantly different with respect to the same characteristics. Edge detection is one of the most frequently used techniques in digital image processing [5, 6, 17, 18, 25]. The boundaries of object surfaces in a scene often lead to oriented localized changes in intensity of an image, called edges.
This observation combined with a commonly held belief that edge detection is the first step in image segmentation, has fueled a long search for a good edge detection algorithm to use in image processing [3, 4]. This search has constituted a principle area of research in low level vision and has led to a steady stream of edge detection algorithms published in the image processing journals over the last two decades. Even recently, new edge detection algorithms are published each year. The following section contains more explanation about edge detection.

2.1 Fundamentals of Edge Detection

Edge detection refers to the process of identifying and locating sharp discontinuities in an image [5, 18, 25]. The discontinuities are abrupt changes in pixel intensity which characterize boundaries of objects in a scene. Classical methods of edge detection involve convolving the image with an operator such as 2-D filter, which is constructed to be sensitive to large gradients in the image while returning values of zero in uniform regions. There is an extremely large number of edge detection operators available, each designed to be sensitive to certain types of edges. Variables involved in the selection of an edge detection operator include

- **Edge orientation**: The geometry of the operator determines a characteristic direction in which it is most sensitive to edges. Operators can be optimized to look for horizontal, vertical, or diagonal edges.

- **Noise environment**: Edge detection is difficult in noisy images, since both the noise and the edges contain high-frequency content. Attempts to reduce the
noise result in blurred and distorted edges. Operators used on noisy images are typically larger in scope, so they can average enough data to discount localized noisy pixels. This results in less accurate localization of the detected edges.

- **Edge structure:** Not all edges involve a step change in intensity. Effects such as refraction or poor focus can result in objects with boundaries defined by a gradual change in intensity. The operator needs to be chosen to be responsive to such a gradual change in those cases. Newer wavelet based techniques actually characterize the nature of the transition for each edge in order to distinguish, for example, edges associated with hair from edges associated with a face.

There are multiple ways to perform edge detection. However, the majority of different methods may be grouped into two categories:

- **Gradient:** The gradient method detects the edges by looking for the maximum and minimum in the first derivative of the image.

- **Laplacian:** The Laplacian method searches for zero crossings in the second derivative of the image to find edges. An edge has the one-dimensional shape of a ramp and calculating the derivative of the image can highlight its location. Suppose we have the following signal, as shown in figure with an edge shown by the jump in intensity.
and if we take the gradient of this signal which, in one dimension, is just the first derivative with respect to $t$ we get the following

![Graph of f(t) showing a gradient]

Clearly, the derivative shows a maximum located at the center of the edge in the original signal. This method of locating an edge is characteristic of the “gradient filter” family of edge detection filters and includes the Sobel method. A pixel location is declared an edge location if the value of the gradient exceeds some threshold. As mentioned before, edges will have higher pixel intensity values than those surrounding it. So once a threshold is set, we can compare the gradient value to the threshold value and detect an edge whenever the threshold is
exceeded. Furthermore, when the first derivative is at a maximum, the second derivative is zero. As a result, another alternative of finding the location of an edge is to locate the zeros in the second derivative. This method is known as the Laplacian and the second derivative of the signal is as shown below:

Edge detection techniques transform images to edge images benefiting from the changes of grey tones in the images. Edges are the sign of lack of continuity, and ending. As a result of this transformation, edge image is obtained without encountering any changes in physical qualities of the main image. Objects consist of numerous parts of different color levels. In an image with different grey levels, despite an obvious change in the grey levels of the object, the shape of the image can be distinguished as shown in Figure 2. 1. When the image intensity abruptly changes from one value to one side of the discontinuity to a different value on the opposite side then it is a step edge. A step edge where the intensity change is not instantaneous but occurs over a finite distance then it is a Ramp edge. If the image intensity abruptly changes value but then returns to the starting value
with in some short distances then it is a Line edge. A ridge edge where the intensity change is not instantaneous but occurs over a finite distance then it is a Roof edge.

![Fig. 2.1 Type of Edges (a) Step Edge (b) Ramp Edge (c) Line Edge (d) Roof Edge](image)

The following figure 2.2 shows the outcome of differentiation function, which is to detect sharp, step and the ramp edges.

![Fig. 2.2 Using differentiation to detect (a) the sharp edges, (c) the step edges with noise, and (e) the ramp edges. (b)(d)(e) are the results of differentiation of (a)(c)(e).](image)
An Edge in an image is a significant local change in the image intensity, usually associated with a discontinuity in either the image intensity or the first derivative of the image intensity. Discontinuities in the image intensity can be either Step edge, where the image intensity abruptly changes from one value on one side of the discontinuity to a different value on the opposite side, or Line Edges, where the image intensity abruptly changes value but then returns to the starting value within some short distance [5, 18, 25]. However, Step and Line edges are rare in real images. Because of low frequency components or the smoothing introduced by most sensing devices, sharp discontinuities rarely exist in real signals. Step edges become Ramp Edges and Line Edges become Roof edges, where intensity changes are not instantaneous but occur over a finite distance. Illustrations of these edge shapes are shown in Figure.2.1.

2.2 Steps in Edge Detection

Edge detection contains three steps namely Filtering, Enhancement and Detection [5, 18, 25]. The overview of the steps in edge detection is as follows

Filtering

Images are often corrupted by random variations in intensity values, called noise. Some common types of noise are salt and pepper noise, impulse noise and Gaussian noise. Salt and pepper noise contains random occurrences of both black and white intensity values. However, there is a trade-off between edge strength and noise reduction. More filtering to reduce noise results in a loss of edge strength [28].

21
**Enhancement**

In order to facilitate the detection of edges, it is essential to determine changes in intensity in the neighborhood of a point. Enhancement emphasizes pixels where there is a significant change in local intensity values and is usually performed by computing the gradient magnitude.

**Detection**

Many pixel points in an image have a nonzero value for the gradient, and not all of these points are edges for a particular application. Therefore, some method should be used to determine which points are edge points. Frequently, thresholding provides the criterion used for edge detection [5, 18, 25, 26, 27].

**2.3 First – Order Derivative Edge Detection**

Most edge detectors are based in some way on measuring the intensity gradient at a point in the image. When we apply function to a gradient operator $\nabla$ then we will get equation 2.1

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

(2.1)

An important quantity in edge detection is the magnitude of this vector, denoted by $\Delta f$, where

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

(2.2)
Another important quantity is the direction of the gradient vector. That is,

$$\text{angle of } \nabla f = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$  \hspace{1cm} (2.3)

Computation of the gradient of an image is based on obtaining the partial derivatives of $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial y}$ at every pixel location. Let the 3x3 area shown in Figure 2.2 represent the gray levels in a neighbourhood of an image. One of the simplest ways to implement a first-order partial derivative at point $z_5$ is to use the following Roberts cross-gradient operators

$$G_x = (z_9 - z_5)$$  \hspace{1cm} (2.4)

and

$$G_y = (z_8 - z_6)$$  \hspace{1cm} (2.5)

These derivatives can be implemented for an entire image by using the masks shown in Figure 2.4 – Figure 2.7 with the procedure of convolution. Another approach of using masks of size 3x3 shown in Figure 2.3 which is given by

$$G_x = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)$$  \hspace{1cm} (2.6)

and

$$G_y = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)$$  \hspace{1cm} (2.7)

A slight variation of these two equations uses a weight of 2 in the centre coefficient:

$$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$  \hspace{1cm} (2.8)

$$G_y = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)$$  \hspace{1cm} (2.9)
A weight value of 2 is used to achieve some smoothing by giving more importance to the center point. Figure 2.4, called the Sobel operators, is used to implement these two equations.

\[
\begin{array}{ccc}
  z_1 & z_2 & z_3 \\
  z_4 & z_5 & z_6 \\
  z_7 & z_8 & z_9 \\
\end{array}
\]

Fig. 2.3 A 3×3 area of an image

\[
\begin{array}{ccc}
  0 & 0 & 0 \\
  0 & -1 & 0 \\
  0 & 0 & 1 \\
\end{array}
\quad
\begin{array}{ccc}
  0 & 0 & 0 \\
  0 & 0 & -1 \\
  0 & 1 & 0 \\
\end{array}
\]

Fig. 2.4 The Roberts operators

\[
\begin{array}{ccc}
  -1 & -1 & -1 \\
  0 & 0 & 0 \\
  1 & 1 & 1 \\
\end{array}
\quad
\begin{array}{ccc}
  -1 & 0 & 1 \\
  -1 & 0 & 1 \\
  -1 & 0 & 1 \\
\end{array}
\]

Fig. 2.5 The Prewitt operators

\[
\begin{array}{ccc}
  -1 & -2 & -1 \\
  0 & 0 & 0 \\
  1 & 2 & 1 \\
\end{array}
\quad
\begin{array}{ccc}
  -1 & 0 & 1 \\
  -2 & 0 & 2 \\
  -1 & 0 & 1 \\
\end{array}
\]

Fig. 2.6 The Sobel operators
2.4 Second – Order Derivative Edge Detection

The Laplacian of a 2-D function \( f(x, y) \) is a second-order derivative defined as

\[
\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}
\]

(2.10)

There are two digital approximations to the Laplacian for a 3×3 region:

\[
\nabla^2 f = 4z_5 - (z_2 + z_4 + z_6 + z_8)
\]

(2.11)

\[
\nabla^2 f = 8z_5 - (z_1 + z_2 + z_3 + z_4 + z_6 + z_7 + z_8 + z_9)
\]

(2.12)

where the \( z \)'s are defined and the Masks for implementing these two equations are shown in Figure. 2.7.

The Laplacian is usually combined with smoothing as a precursor to find edges via zero-crossings. The 2-D Gaussian function is as shown in Equation 2.13.
\[ h(x, y) = -e^{-\frac{x^2 + y^2}{2\sigma^2}} \]

where \( \sigma \) is the standard deviation, blurs the image with the degree of blurring being determined by the value of \( \sigma \). The Laplacian of \( h \) is

\[ \nabla^2 h(x, y) = -\left[ \frac{x^2 + y^2 - 2\sigma^2}{\sigma^4} \right] e^{-\frac{x^2}{2\sigma^2}} \]

This function is commonly referred to as the **Laplacian Of Gaussian (LOG)**.

![LOG Function](image)

**Fig. 2.9 Dimension coordinate of Laplacian of Gaussian (LOG)**

After calculating the two-dimensional second-order derivative of an image, we find the value of a point which is greater than a specified threshold and one of its neighbours is less than the negative of the threshold. The property of this point is called zero-crossing and we can denote it as an edge point. We note two additional properties of the second derivative around an edge.
(1) It produces two values for every edge in an image (an undesirable feature)

(2) an imaginary straight line joining the extreme positive and negative values of the second derivative would cross zero near the midpoint of the edge.

This zero-crossing property of the second derivative is quite useful for locating the centres of thick edges.