**Chapter 5**

**Image Processing Techniques**

**5.1 Introduction**

Image Processing is the breakdown of images or video frames in digital format to abstract useful data from them. In the case of this research, images are processed to abstract features for hand shape recognition.

These are image processing techniques such as edge detection, background subtraction using Gaussian Mixture Models, adaptive skin detection, connected components analysis. Each of these techniques is discussed in a separate subcategory below.

**5.1 Edge Detection**

In image analysis, edge detection is one of the most well-known and commonly used operations. In image processing relating to computer vision, the edge detection uses the localization of important variations of a gray level image by the detection of the physical and geometrical characteristics of objects of the scene. Hence, it is an elementary process that identifies and figure out an article and boundaries among objects and the background in the image. Edge detection frequently used in numerous application such as medical image processing, biometrics and many more by using object detection. Edge detection is an active area of study as it facilitates higher level image processing.

Edges played an important role in local changes in the image and that’s the reason are important tool for analyzing images. Edges primarily appears on the boundary between two different sources in an image, nevertheless, an edge is defined by a disjointedness in gray level values. Figure 5.1 shows the disjointedness in gray level values. An edge can also be expanded as a boundary between an object and the background. The structure of edges in images rely on numerous aspects: The geometrical and optical properties of the object, the illumination conditions, and the noise level in the images [11].



*Figure 5.1: Edge intensity example, Source: Gonzalez & Woods, Digital Image Processing (2002)*

Discontinuities in the image intensity can be demonstrate in the form of (1) step discontinuities and (2) line discontinuities. However, in first one the image intensity abruptly changes from one value on one side of the discontinuity to a different value on the opposite side, where as in the later one, the image intensity abruptly changes value but then returns to the starting value within some short distance, hence, step and line edges are few in real images. Sharp discontinuities are rarely existing in real world signals due to the low-frequency components or the smoothing introduced by most sensing devices. Step edges converted to ramp edges, whereas line edges become roof edges, where intensity changes are not instantaneous but occur over a finite distance. Occasionally, edges could have features of both step and line. Additionally, spatial masks can be used to detect all types of discontinuities in an image. Figure 5.2 illustrates different types of edge discontinuities.

****

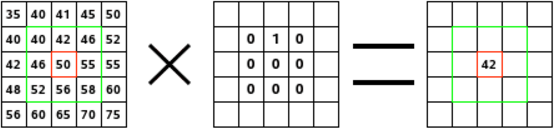
*Figure 5.2: One-dimension Edge profiles, Source: Jain* et al, *Machine Vision, chapter- 5 [13]*

Edge detecting in an image considerably reduces the amount of data and filters out unusable information, whereas preserve the vital structural properties in an image.

**5.1.1 Convolution**

Filtering is a common process in digital images to remove the noise. Convolution is a general filter effect for image and modifies the spatial characteristics of the image. It is done by a simple mathematical operation by multiplying a pixel’s and its neighboring pixel values by a matrix called Kernel. Kernel or Mask is a small integer matrix, usually of size 3x3, used for convolution. Convolution is done by moving the kernel on the image, starting from the top left corner of the image to the bottom right within the boundaries. With each pixel of the image, the kernel cell value is multiplied and added together. Mathematically it can be written as:

As a result, the output is a new improved filtered image. The choice of the kernel affects the results accordingly. Figure 5.3 shows the process of convolution using 3x3 kernel.



*Figure 5.3: Convolution example, Image (left) Kernel (middle) output Image (right),*

*Source:* *docs.gimp.org*

**5.2 Steps for Edge Detection**

Edge detection’s algorithms consist of steps as under;

* **Filtering**

Filtering is a process in which noise is minimized, however, noise is reduce as much as further do not affect the meaningful edges, hence, increasing filtering further will lead to loss of edge strength.

* **Enhancement**

To determine a change in intensity, the neighborhood of a point is vital, nevertheless, enhancement focuses on pixels, where a significant change in localized intensity values are noticed, which is commonly executed by computing the gradient magnitude.

* **Detection**

Detection is used to identified the point belongs to edge points as the points with only strong edge content are needed in the process, however, many points in an image have a non-zero value for the gradient, and not all of these points are edges for a particular application. Frequently, thresholding provides the criterion used for detection.

* **Localization**

The location of the edge can be estimated with sub-pixel resolution. It can be decided which of the local maxima output by the filter are meaningful edges and which are caused by noise.

**5.3 Edge Detection Techniques**

Over the last three decades edge detection techniques have been studied thoroughly and various literature is available on these techniques. The prominent characteristic of edge detection technique is its capability to extract the accurate edge line with an appropriate alignment. Moreover, the limitation has been noticed is that there is not yet any common performance directory to evaluate the functioning of the edge detection techniques, whereas the performance of an edge detection techniques are mostly required a personal analysis and separately dependent to its application.

Edge detection methods could convert original images into edge images benefits from the changes of grey tones in the image and it has capability to review image, nonetheless been used by advanced computer vision algorithms.

Various literature on edge detection techniques are available, hence, the most commonly used discontinuity based edge detection techniques are reviewed in this section some of those techniques are Roberts edge detection, Sobel Edge Detection, Prewitt edge detection, Kirsh edge detection, Robinson edge detection, Marr-Hildreth edge detection, LoG edge detection and Canny Edge Detection, are discussed in detail;

**5.3.1 Roberts Edge Detection**

The Roberts edge detection was introduced by Lawrence Roberts (1965) [35]. This method performs a simple and fast 2D spatial gradient measurement on an image. This technique focuses on high spatial frequency regions as they often correspond to edges. A grayscale image is the input to the operator and the same as to the output. In the output, pixel values in every point represents the estimated complete magnitude of the spatial gradient at that point for the input image.

Gx and Gy are calculated using mask shows in Table 3.1

Table 3.1: Masks used in Robert’s operator

|  |  |
| --- | --- |
| -1 | 0 |
| 0 | 1 |

|  |  |
| --- | --- |
| 0 | -1 |
| +1 | 0 |

Gx Gy

**5.3.2 Sobel Edge Detection**

The Sobel edge detection method was presented by Sobel in 1970 [14]. This method of edge detection for image segmentation finds edges using the Sobel approximation to the derivative. It starts looking for the edges at those points where the gradient is highest. The Sobel method performs a 2D spatial gradient quantity on an image and so emphasizes the areas of high spatial frequency that correspond to edges. This method is used to find the estimated absolute gradient magnitude in an input grayscale image at each point. In inference at least the operator consists of a pair of 3x3 complication kernels as shown in under Table 52. One kernel is simple and the other rotated by 90o. It is very similar to the Roberts Cross operator.

Table 5.2: Masks used in Sobel’s operator

|  |  |  |
| --- | --- | --- |
| -1 | -2 | -1 |
| 0 | 0 | 0 |
| +1 | +2 | +1 |

|  |  |  |
| --- | --- | --- |
| -1 | 0 | +1 |
| -2 | 0 | 2 |
| -1 | 0 | +1 |

Gx  Gy

**5.3.3 Prewitt Edge Detection**

The Prewitt edge detection was represented by Prewitt in 1970 [14]. Prewitt method is a correct way to estimate the magnitude and orientation of an edge. To estimate the direction from the magnitudes in the x and y-directions different gradient edge detection needs a quiet time consuming calculation, the compass edge detection gets the direction directly from the kernel with the maximum rejoinder. It is limited to 8 possible directions; though knowledge demonstrates that most direct direction estimates are not too perfect. This edge detector is estimated in the 3x3 neighborhood for eight directions. All the eight convolution masks are calculated. One complication mask, with the largest module is then selected. Table 5.3 shows the masks used in Prewitt’s operator.

Table 5.3: Masks used in Prewitt’s operator

|  |  |  |
| --- | --- | --- |
| -1 | -1 | -1 |
| 0 | 0 | 0 |
| +1 | +1 | +1 |

|  |  |  |
| --- | --- | --- |
| -1 | 0 | +1 |
| -1 | 0 | +1 |
| -1 | 0 | +1 |

Gx Gy

Based on the observation it has been identified that Prewitt detection is relevantly easier to apply computationally than the Sobel detection, but it is more prone to produce noisier outcome.

**5.3.4 Kirsch Edge Detection**

Kirsch edge detection was proposed by Kirsch (1971) [36]. In this technique the masks are defined

by considering a single mask and then rotating it to eight main compass directions: North, West, South, East, Northwest, Southwest, Southeast and Northeast. The masks are distinctive as follows

N, W, S, E, NW, SW, SE and NE respectively.

l1 l2 l3 l4

l5  l6  l7 l8

The maximum value found by convolution of each mask with the image is defined as edge magnitude. The mask that generates the maximum magnitude defines the direction. For example, mask l6 corresponds to a diagonal edge whereas mask l1 corresponds to a vertical edge. It can be notice that the last four masks are actually the same as the first four, but flipped about a central axis.

**5.3.5 Robinson Edge Detection**

The Robinson method was represented by Robinson in 1977, which is same as to Kirsch masks but is easier to implement as they depend only on coefficients of 0, 1 and 2 [37]. The masks are symmetrical about their directional axis, the axis with the zeros. Only four masks are computed and then the rest of the masks are computed by negating the result from the first four. The masks are as follows:

l1 l2 l3 l4

l5  l6  l7 l8

The maximum value obtained from applying all eight masks to the pixel neighborhood is defined as magnitude of the gradient. The angle of the line of zeroes in the mask yielding the maximum response can be approximated as the angle of the gradient.

**5.3.6 Marr-Hildreth Edge Detection**

The Marr-Hildreth was proposed in 1980, which is a technique of detecting edges in digital images that is continuous curves where there are well-built and fast distinctions in image brightness [38]. It is easy and operates by convolving the image with the LoG function. Later, to find the edges, the zero-crossings are revealed in the filtered result. The LoG method, due to its image shape while turned up-side-down is also referred as the Mexican hat wavelet. The Marr-Hildreth edge detector algorithm is:

Smooth the image using a Gaussian

• A two-dimensional Laplacian is applied to the smoothed image (often the first two steps are combined into a single operation)

• Look for sign changes by applying loop through the result. If there is a sign change plus the slope across, mark as an edge if the sign change is greater than some threshold.

• It is possible to run the result of the Laplacian through a hysteresis alike to Canny’s edge detection to get better results, although this is not how the edge detector was originally implemented.

**5.3.7 LoG Edge Detection**

The Laplacian of Gaussian (LoG) was introduced by Marr (1982). The LoG of an image f(x,y) is a second order derivative defined as,

This method has two effects, on one hand it smooths the image and on other hand computes the Laplacian, which produces a double edge image. Locating edges then consists of finding the zero crossings between the double edges. Generally mask as shown in Table 5.4 is used to implement Laplacian function,

|  |  |  |
| --- | --- | --- |
| 0 | -1 | 0 |
| -1 | 4 | -1 |
| 0 | -1 | 0 |

|  |  |  |
| --- | --- | --- |
| -1 | -1 | -1 |
| -1 | 8 | -1 |
| -1 | -1 | -1 |

Table 5.4: Masks used in LoG s operator

GxGy

The Laplacian is generally used to found whether a pixel is on the dark or light side of an edge.

**5.3.8 Canny Edge Detection**

In commercial world, the Canny edge detection technique is considered as one of the standard edge detection techniques. Initially it was introduced by John Canny for his Master’s thesis at MIT in 1983, and it is still excel many of the newer algorithms that have been developed over the years [12]. Canny is a highly used method to identify the edges by separating noise from the image before finding edges of image. The same theory is a better solution that without disturbing the features of the edges in the image afterwards it applying the tendency to find the edges and the serious value for threshold.

**1. Detection:** In detection the chances of identifying the real edge point should be increased, however, the chances of unreal and default non-edge points is decreased. This feature leads to get optimal signal-to-noise ratio.

**2. Localization:** The detected edges should be as nearer to the real edges.

**3. Number of responses:** One real edge should not produce more than one detected edge (it has been noticed that this is an absolute requirement).

The algorithm takes place in 5 separate steps:

* Smoothing: To remove noise, blurring of the image.
* Finding gradients: The gradients of the image having large magnitudes should be marked as edge.
* Non-maximum suppression: Only local maxima should be marked as edges.
* Double thresholding: Apply thresholding to determine potential edges.
* Edge tracking by hysteresis: All the other edges that are not connected to a very strong edge are suppressed to determine final edges.

All the steps are described in detail in the subsequent subsections.

**5.3.8.1 Smoothing**

It is predetermined that all images taken from a camera will encompass of some amount of noise, however, to prevent noise for edges it is necessary to reduce the noise. For that reason, a Gaussian filter is first applied to smooth the image. The kernel of a Gaussian filter with a standard deviation of ϭ = 1.4 is given in equation (1).

(1)

**5.3.8.2 Finding Gradients**

One of the prominent feature of Canny algorithm is finding those edges, where the intensity of the grayscale images changes the most. With the help of gradients of the image these areas can be determine. Sobel-operator is applied on the smoothed image to determine the gradients at each pixel. In the first step, the kernels are applied to approximate the gradient in the x- and y-direction respectively, can see in equation (2).

(2)

By applying the law of Pythagoras, the gradient magnitudes can then be determined as a Euclidean distance measure as shown in equation (3). To reduce the computational complexity sometimes Manhattan distance measure also can be applied as shown in equation (4).

(3)

(4)

Gx and Gy are the gradients in the x- and y-directions respectively.

At this step, an image of the gradient magnitudes often indicate the edges prominently. However, the edges are eventually wide and couldn’t stipulate the definite area of edges. Besides to make it possible to evaluate this, the direction of the edges must be determined and gathered as indicated in equation (5).

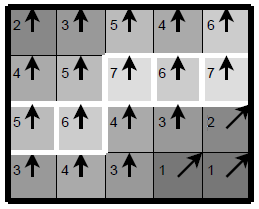
(5)

**5.3.8.3 Non-Maximum Suppression**

Non-maximum suppression is used to convert the “blurred” edges in the image of the gradient magnitudes to “sharp” edges. Basically the results have been determined by preserving all local maxima in the gradient image, whereas deleting the rest. The algorithm is for each pixel in the gradient image:

* Round the gradient direction θ to nearest 45◦, corresponding to the use of an 8-connected neighborhood.
* Current pixel’s edge strength is compared with the edge strength of the pixel in the positive and negative gradient direction. i.e. compare with the pixels to the north and south if the gradient direction is north (theta = 90◦).
* Preserve the value of the edge strength if the edge strength of the current pixel is largest.

Suppress (i.e. remove) the value if it is not the largest.

**

*Figure 5.3: Simple model of non-maximum suppression*

Over and above, Figure 5.3 illustrates a simple model of non-maximum suppression. Majority of the pixels have gradient directions pointing to the north, therefore compared with the pixels above and below. The white borders represents the pixels that turn out to be maximal in this comparison, whereas rest of the pixels will be suppressed.

**5.3.8.4 Double Thresholding**

Following the non-maximum suppression step, the remaining edge-pixel are still noticeable with their strength pixel-by-pixel. It is assumed that most of them are commonly the actual edge in the image but few could be created by noise and difference in color for example due to uneven surfaces. To overcome this issue, the easiest approach to recognize between these would be to use a threshold, so that only edges stronger that a certain value would be preserved. The Canny edge detection algorithm uses double thresholding. Edge pixels stronger than the high threshold are marked as strong; edge pixels weaker than the low threshold are suppressed and edge pixels between the two thresholds are marked as weak.

**5.3.8.5 Edge Tracking by Hysteresis**

The strong edges are explicated as “certain edges”, and can immediately be included in the final edge image. On contrary, the weak edges are included if and only if they are connected to strong edges. The phenomena behind is that noise and other minor elements are unlikely lead to strong edge (with proper adjustment of the threshold levels). Thus, in the original image, the strong edges will (almost) only be due to true edges, however, the weak edges can either be due to true edges or noise/color variations. The latter type will probably be distributed independently of edges on the entire image, and thus only a small amount will be located adjacent to strong edges. Weak edges due to true edges are much more likely to be connected directly to strong edges.

BLOB-analysis (Binary Large OBject) can be used for edge tracking to be implemented. The edge pixels are distributed into connected BLOB’s using 8-connected neighborhood. BLOB’s containing at least one strong edge pixel are then preserved, while rest of the BLOB’s are suppressed.

**5.4 Analysis of Edge Detection Methods**

Roberts edge detector, Sobel Edge Detector, Prewitt edge detector, Kirsch, Robinson, Marr-Hildreth edge detector, LoG edge detector and Canny Edge Detector are the few prominent relative performance edge detection techniques, which were under discussion in this chapter.

Based on the literature review, Roberts, Sobel, Prewitt, Kirsch and Robinson algorithms

|  |
| --- |
|  |
| Original Roberts Sobel |
|  |
| Prewitt Robinson Kirsch |
|  |
| Marr-Hildreth LoG Canny |

*Figure 5.4: Different Edge detection methods, Source: M.Radha2 et al, 2011*, *Edge detection techniques for* *Image segmentation [15]*

are simple to implement and can detect edges and their orientations but have major drawbacks in sensitive to noise. As the dimension of the kernel filter and its coefficients are static and it cannot be adapted to a given image thus the results might be inaccurate. Marr-Hildreth and LoG algorithms find the correct edges as they test wider area around the pixel but they malfunction at corners, curves and where the gray level intensity fluctuates.

Figure 5.4 demonstrates that Roberts, Sobel and Prewitt outcomes are actually deviated from the others, in contrast, Marr-Hildreth, LoG and Canny develop almost the same edge map. In addition, Kirsch and Robinson edge maps look similar. Canny demonstrates the best results among all of them.

The performance of the Canny’s algorithm mainly depends on the changing parameters which are standard deviation for the Gaussian filter, and its threshold values. Unlike Roberts and Sobel, the Canny’s operation is not very susceptible to noise. By applying smoothing concept, the finding of errors is effective by using the probability. Improving signal with respect to noise ratio. Canny edge detection algorithm is more costly due to complex calculations in comparing to Sobel, Prewitt and Robert’s operator. If the Canny detector worked well it would be superior.

Even though, so many edge detection techniques are available in the literature, it is a challenging task to the research communities to detect the exact image without noise from the original image.