AUTONOMOUS ROBOT FOLLOWING HUMAN
(VISION PART)

Su Shyn Sern, Nicholas

Bachelor in Mechatronics Engineering

2009

"I hereby declared that I have read through this report and found that it has comply the partial fulfillment for awarding the degree of Bachelor of Mechatronics Engineering

Signature : ....

Supervisor's Name : HAIROL NIZAM BIN MOHD SHAH

Date : 13/05/09

AUTONOMOUS MOBILE ROBOT FOLLOWING HUMAN (VISION PART)

SU SHYN SERN, NICHOLAS

This Report Is Submitted In Partial Fulfillment Of Requirements For The Degree of Bachelor In Mechatronics engineering

Faculty of Electrical Engineering
Universiti Teknikal Malaysia Melaka

April 2009

"I declare that this report entitle "AUTONOMOUS MOBILE ROBOT FOLLOWING HUMAN (VISION PART)" is the result of my own research except as cited in the references. The report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree."

Signature : ....
Name : Su Shyn Sern, Nicholas
Date : 13/05/2009

ACKNOWLEDGEMENTS

This project could not have been completed without En. Hairol Nizam bin Mohd Shah who not only served as my supervisor but also encouraged and challenged me throughout my academic program. And to my panels, Sulaiman b. Sabikan and Muhammad Herman b. Jamaluddin, thank you for evaluating my report with utmost fairness and objectivity. I would also like to express my gratitude to Tam You Wei for his cooperation in completing this project, and to all my other friends who provided advice and guidance. And last but not least, to my parents for continuing supporting me and my girlfriend for being there for me when I need her,

ABSTRACT

This project will detail out the process of developing the vision part of a human following robot. Matlab will be used to develop a system consisting of a classification algorithm and a tracking algorithm. This system will allow the robot to differentiate a human in a picture. This project will utilize a wireless camera to allow greater mobility. A friendly graphic user interface will also be developed to monitor the process of the system.

ABSTRAK

Projek ini akan menerangkan process membina sebuah perisian robot untuk mengikut perjalanan seorang manusia. Perisian Matlab akan digunakan untuk membina perisian ini yang terdiri daripada algorithm pengkelasan and algorithm menjejak. SIstem ini akan membenarkan robot ini mengenali seorang manusia dalam sebuah gambar. Projek ini akan menggunakan sebuah camera tanpa wayar untuk mobiliti yang tinggi.

TABLE OF CONTENT

ABSTRACT iv

ABSTRAK v

TABLE OF CONTENT vi

CHAPTER 1: INTRODUCTION 1

1.1 Project Overview 1

1.2 Problem Statement 1

1.3 Objectives 1

1.4 Scope 2

1.5 Literature Review 2

CHAPTER 2:METHODOLOGY 4

2.1 Part 1: Develop detection and tracking algorithm 5

2.2 Part 2: Interfacing the Camera with the Program 6

2.3 Part 3: Combining the Algorithm with the Mobile Robot Control 7

2.4 Image Processing Methods used 7

2.4.1 Edge Detection 7

2.4.2 Thresholding 10

2.4.3 Morphological Processing 12

2.5 Matlab's Image Processing Toolbox and Image Acquisition Toolbox 14

2.6 The Human Detection and Tracking Algorithm 15

CHAPTER 3:RESULTS 18

3.1 Experimental Setup 18

3.2 Results Part 1: Human detection 20

3.2.1 Edge detection for segmentation 21

3.2.2 Thresholding for segmentation 23

3.2.3 Processing time of segmentation 26

3.3 Results Part 2: Tracking and GUI 27

3.4 Results Part 3: Completed human following robot and its GUI. 28

CHAPTER 4:DISCUSSION AND SUGGESTION 29

4.1 Discussion 29

4.2 Suggestion 31
CHAPTER 5:CONCLUSION 32
REFERENCE 33
APPENDIX

LIST OF FIGURES

Figure 2.0: Project Flowchart 5
Figure 2.1 Detection and tracking algorithm flowchart 6
Figure 2.2 Wireless camera and receiver 7
Figure 2.3 RG59-to-USB converter 7
Figure 2.4: Signal containing an edge 8
Figure 2.5: First derivative of the signal 8
Figure 2.6: Second derivative of the signal 9
Figure 2.7: Sliding of mask and calculation of new pixel. 9
Figure 2.8: Edge detection operators 10
Figure 2.9: Example of thresholding 11
Figure 2.10: Example of thresholding 11
Figure 2.11: Dilation of set A by a square structuring element B 12
Figure 2.12: Erosion of set A by a square structuring element B 13
Figure 2.13: effect of applying boundary extraction 13
Figure 2.14: Result of region filling. 14
Figure 2.15: Position of robot from human 16
Figure 2.16: Tracking 17
Figure 3.0: Matlab interface 18
Figure 3.1: Receiver setup 19
Figure 3.2: Wireless camera setup 19
Figure 3.3: Area used for background of the images 20
Figure 3.4: Area used for testing tracking algorithm 20
Figure 3.5: Test pictures 21
Figure 3.6: Segmentation of 'testpic1.jpg 'using Roberts edge detection 22
Figure 3.7: Segmentation of 'testpic2.jpg 'using Roberts edge detection 22
Figure 3.8: Segmentation of 'testpic.3.jpg 'using Roberts edge detection 23
Figure 3.9: Segmentation of 'testpic4.jpg 'using Roberts edge detection 23
Figure 3.10: Segmentation of 'testpic1.jpg' using global thresholding 24

Figure 3.11: Segmentation of ‘testpic2.jpg’ using global thresholding............... 24
Figure 3.12: Segmentation of ‘testpic2.jpg’ using threshold = 0.2 25
Figure 3.13: Segmentation of ‘testpic3.jpg’ using global thresholding............... 25
Figure 3.14: Segmentation of ‘testpic4.jpg’ using global threshold 26
Figure 3.15: Segmentation of ‘testpic4.jpg’ using threshold = 0.2 26
Figure 3.16: GUI for real time human detection and tracking............................ 28
Figure 3.17: The robot following a human and its GUI 28

LIST OF TABLES

Table 2.0: Functions used to develop human detection and tracking algorithm. 14
Table 3.0: Process time of each method 27

CHAPTER 1

INTRODUCTION

1.1 Project Overview

In this project, a human tracking algorithm will be developed to allow a mobile robot to follow a human. A wireless camera will be used for image capturing, and Matlab will be use to process the image captured, followed by controlling the mobile robot to follow the human.

1.2 Problem Statement

Currently in the library, librarians have to push a cart with many books when reorganizing them. This can be a burden, especially if the librarian is a senior citizen. An autonomous robot that follows humans will help solve the problem. Therefore, I propose to develop the vision part of an autonomous human following robot that is able to identify a human being and follow it.

1.3 Objectives

The objective of this project is to:

1. To developed a classification and tracking algorithm using Matlab's image acquisition and image processing library.
2. To build a graphic user interface (GUI) for the program.
3. To interface a wireless camera with the program for real-time image processing.

4. To combine this algorithm with the mobile robot control program to produce a human following mobile robot.

1.4 Scope

For this project, only the vision software part will be studied and developed. After the human tracking algorithm is completed, it will be combined with a completed robot control program. The limitation of this project is that it can work in a controlled background, basically white in colour.

1.5 Literature Review

Human detection and tracking is currently a hot debated topic in the research of computer vision. The literature of human tracking is too vast to be discussed in detail, therefore only a few of those literature will be reviewed to determine the best possible human tracking algorithm.

In [5], an algorithm was developed that tracks a human by building a model of appearances from clustering of candidate body segments without comparing with human motion dynamics. Comparing with human motion dynamics is unreliable because humans can move very fast. Frame to frame comparison will show a great deal of acceleration, which will likely be deem as inhuman motion by the comparing algorithm. To solve this, they compare by appearances, because the appearance of a humans does not tend to change between frame, such as color and texture of clothing. In their algorithm, they have managed to include the ability to track multiple humans, and also recover when it loses track, for example when a car passes by and blocks the view to the human.

In [4], we see the combination of the two main categories of human tracking, which are shaped-based human detector and motion-based human detector. Shaped-based human detectors suffer from 2 drawbacks; high false detection rates and slow processing time. This is caused because it has to process the whole image at different scale to find human figures. Motion-based human detectors has lower false detection rate because they analyze the motion pattern of an object and compare it to human motion dynamic. The weakness is that it has to detect an object first. By combining these two categories, the shaped-based human detector will detect possible human candidates, then tracked for a sufficient amount of time and finally verified with human motion dynamics.

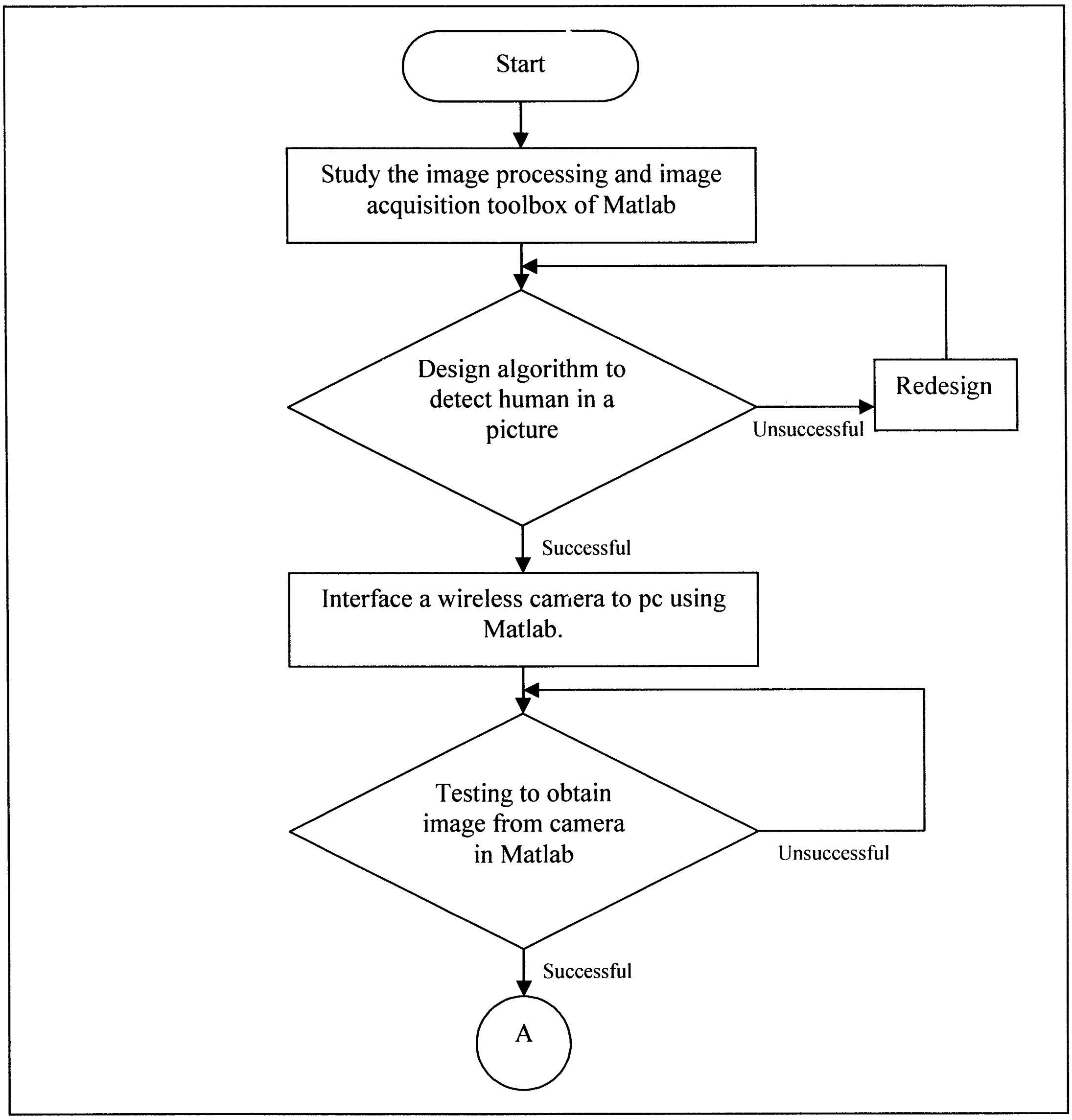
In article [2], a method called Distance Transform is used for shaped- based detection on a moving vehicle. This will be crucial in this project for this deals with what happens when the camera gets nearer to an object of interest. The size will gradually increase as the camera gets closer to the object, causing difference when comparing to existing models of the object.

There is much more to be studied to develop a reliable human detection and tracking algorithm.

CHAPTER 2

METHODOLOGY

This chapter discusses the approach taken to complete the project. Figure 3.1 shows the flow chart of project.



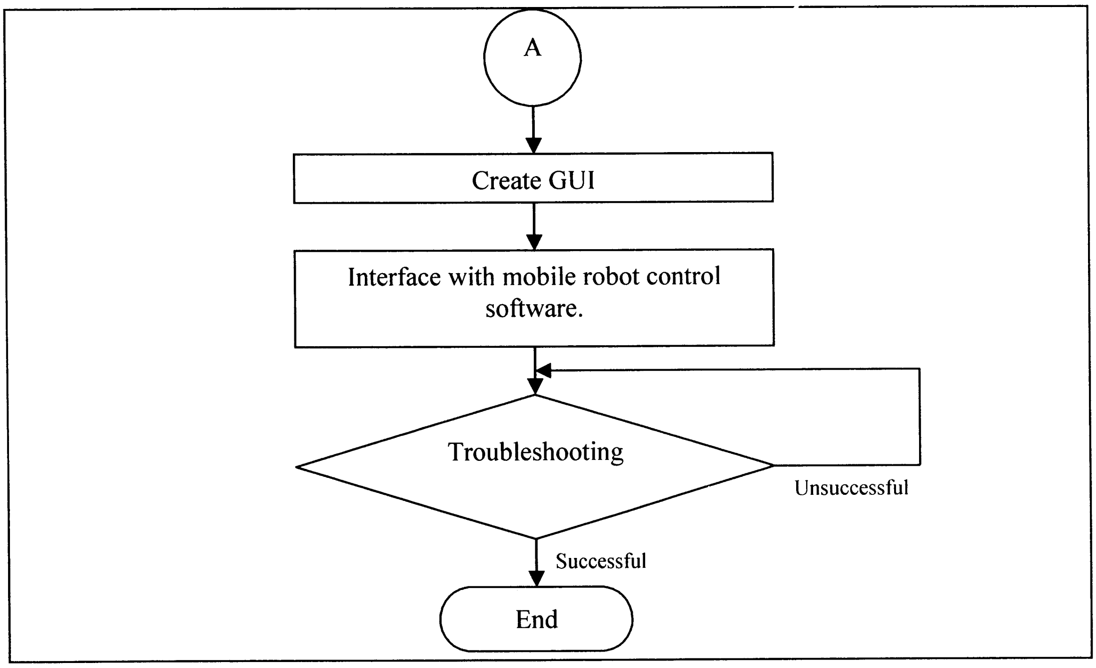


Figure 2.0: Project Flowchart

Basically, there will be three parts in implementing this project. The first part will be to develop the detection and tracking algorithm. The second part involves interfacing the camera with the program to obtain real time processing. The third part will be troubleshooting the algorithm combined with the mobile robot control software.

2.1 Part 1:Develop detection and tracking algorithm

The first step in the algorithm is to recognize objects in an image captured by the camera. Then, the algorithm must determine whether the object segmented is human or not. After determining the object is human, the centroid of the human is determined. This centroid is then compared with the center of the image to get the location of the human with respect to the camera, either at the left or right of the camera. If the human is not in the center of the camera view, then corrective measures is taken so that the human will be in the center of the camera view. Figure

2.1 shows the flowchart for the detection and tracking algorithm. A more completed explanation of the algorithm will be provided in Section 2.6.

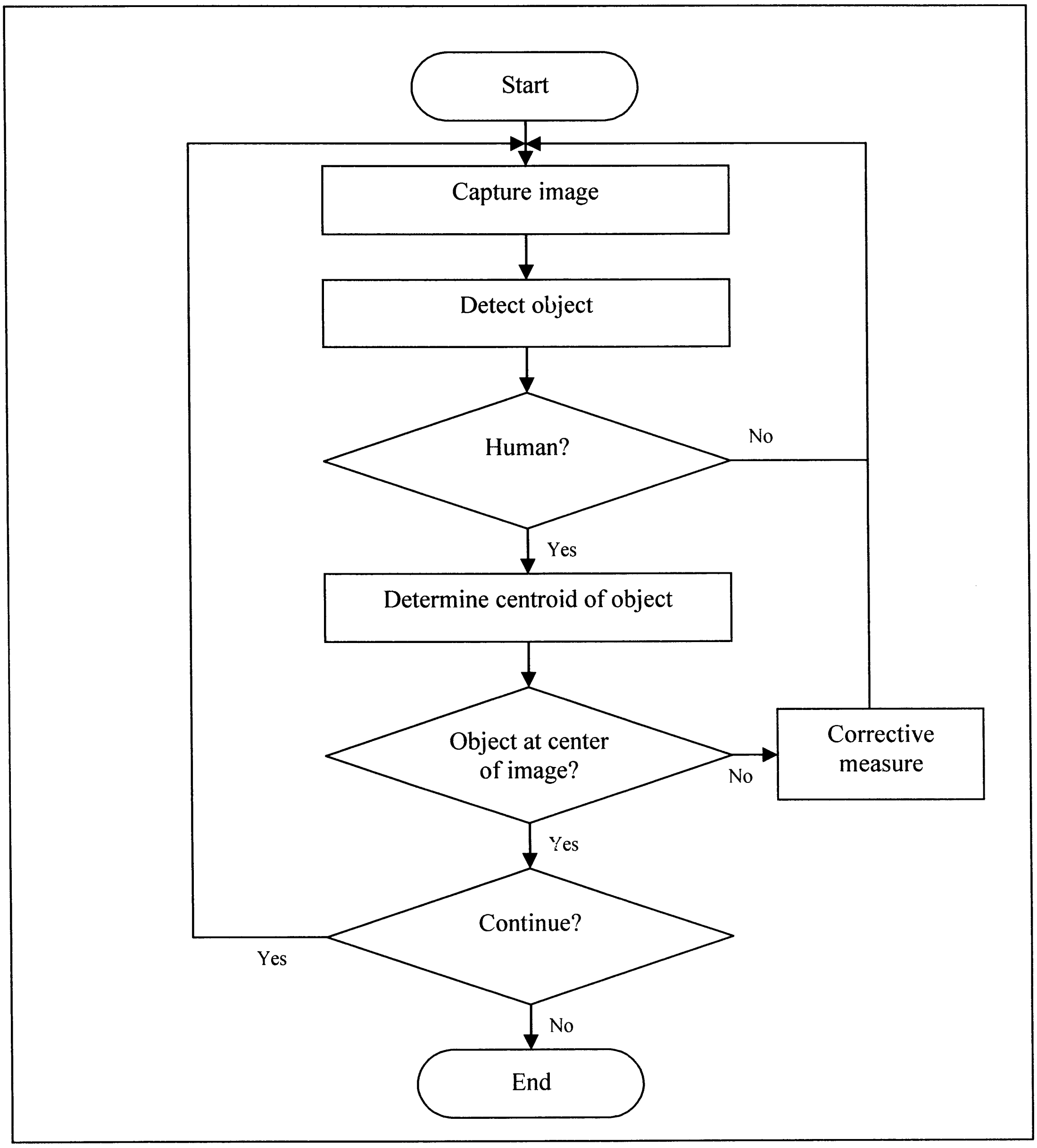


Figure 2.1 Detection and tracking algorithm flowchart

2.2 Part 2: Interfacing the Camera with the Program

For the purpose of this project, a wireless camera (Figure 3.3) will be used for image acquisition. Matlab itself contain functions to detect and communicate with any camera and their drivers in Windows through the USB ports. The wireless camera's transmitter uses the normal RG 59 video cable; therefore, a RG 59-to-USB converter (Figure 3.4) is used. Once interfacing was achieved, a GUI was developed to support a live video feed. Then, the detection and tracking algorithm is applied to the video feed via frames or snapshots obtained from the video feed. Upon study, it was learnt that a live video feed in Matlab is actually the display of captured images on an axis at very fast speed.

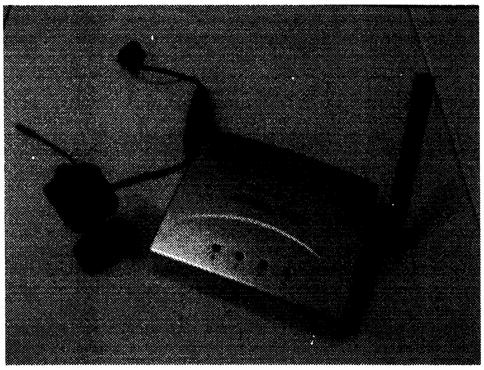


Figure 2.2 Wireless camera and receiver

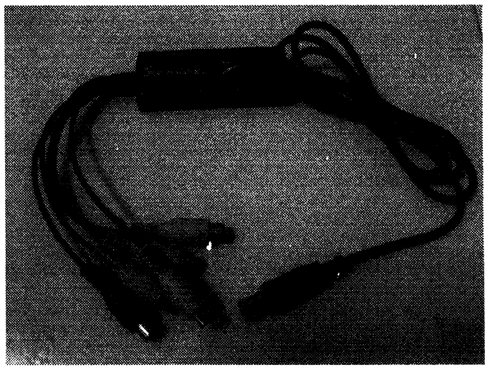


Figure 2.3 RG59-to-USB converter

2.3 Part 3: Combining the Algorithm with the Mobile Robot Control

The corrective measure mentioned in section 3.1 is the mobile robot control software. Once the algorithm is able to detect and track a human, it is merged with the mobile robot control software to enable the mobile robot to position itself so that the human will be centered and distanced in the camera view, in other words making the robot track the human.

2.4 Image Processing Methods used

In order to explain the algorithm, the image processing methods used must be explained in order for any reader to have a better understanding.

2.4.1 Edge Detection

Edge detection is used to detect points in a digital image which experience a sharp change in brightness. The lines obtain can be used for feature detection and feature extraction. Many types of mathematical models have been developed for edge detection, such as the Canny operator, Sobel operator, Prewitt operate and many more. Each of these methods uses a different approach in obtaining lines in an image, but can be grouped into two categories, gradient and Laplacian. The gradient method detects the edges by looking for the maximum and minimum in the first derivative of the image. The Laplacian method searches for zero crossings in the second derivative of the image to find edges.

Suppose we have a signal as in Figure 2.5, with an edge shown by the jump in intensity.

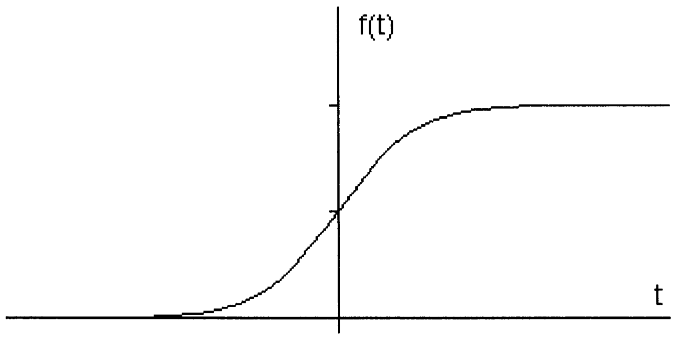


Figure 2.4: Signal containing an edge

If we take the gradient of this signal (which, in one dimension, is just the first derivative with respect to t) we get the signal in Figure 2.6.

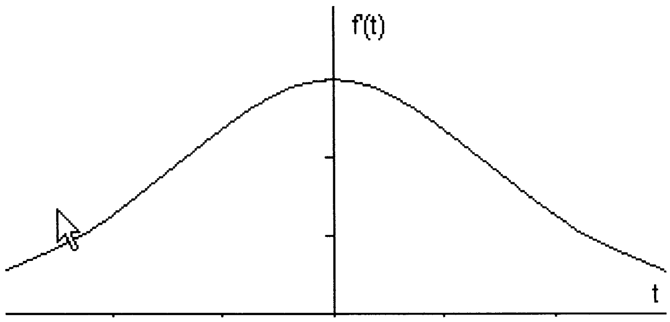


Figure 2.5: First derivative of the signal

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, you 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 to 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 shown in Figure 2.7.

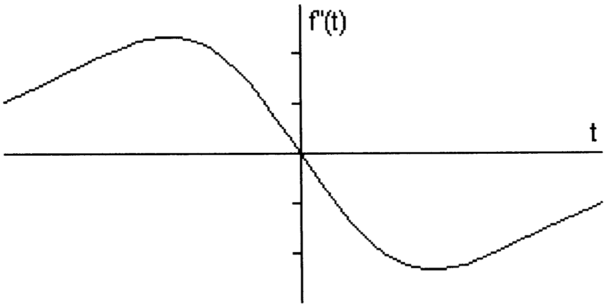
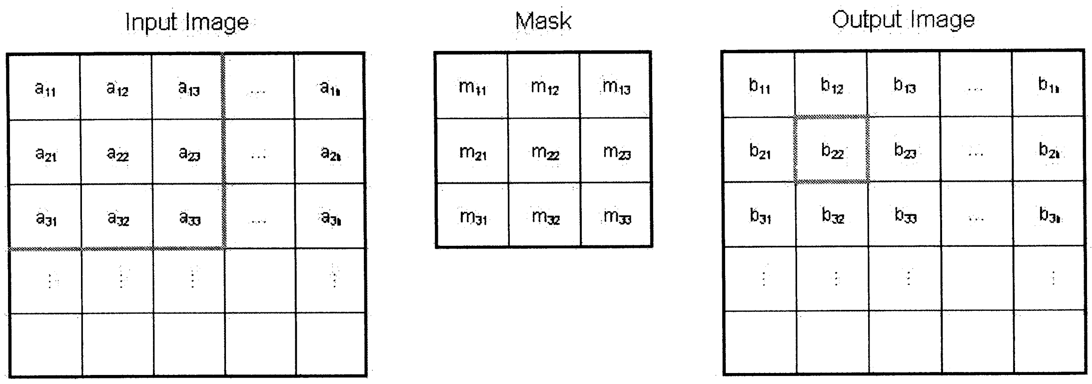


Figure 2.6: Second derivative of the signal

The technique required for edge detection is normally sliding a variety of mask over the image, changes that pixel's value and then shifts one pixel to the right and continues to the right until it reaches the end of a row. It then starts at the beginning of the next row. Figure 2.7 shows the process.



$b_{22} = (a_{11}*m_{11}) + (a_{12}*m_{12}) + (a_{13}*m_{13}) + (a_{21}*m_{21}) + (a_{21}*m_{21}) + (a22*m_{22}) + (a_{23}*m_{23}) + (a_{31}*m_{31}) + (a_{32}*m_{32}) + (a_{33}*m_{33})$

Figure 2.7: Sliding of mask and calculation of new pixel.

The type of mask depends on the operator required. Figure 2.8 shows a few types of masks.

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

Gx

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

Gy

(a) Sobel operator

+1	0
0	-1

Gx

0	+1
-1	0

Gy

(b) Roberts oprator

0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

−1	2	−1
2	−4	2
−1	2	−1

(c) Prewitts operator

Figure 2.8: Edge detection operators

2.4.2 Thresholding

A simple and reliable method for image segmentation, thresholding is used to convert grayscale images to binary images by setting pixels with intensities values below or above the threshold value to black or white respectively. Figure 2.9 show a visual example of thresholding.

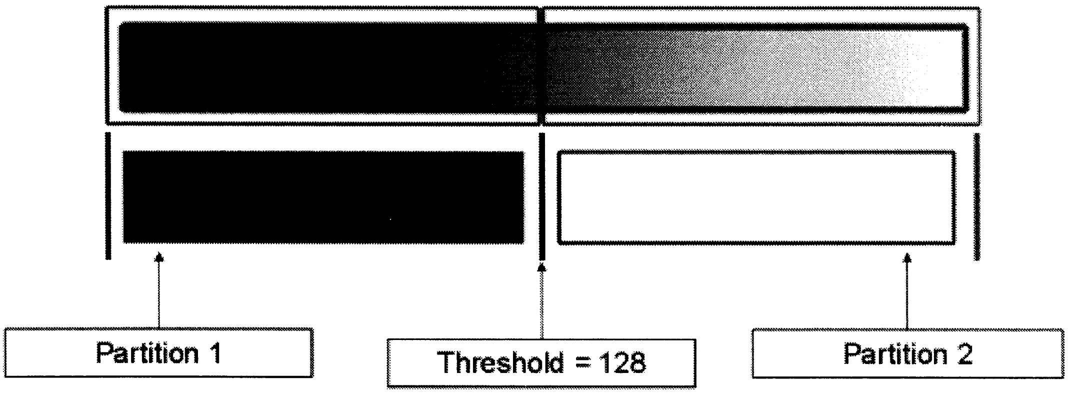


Figure 2.9: Example of thresholding

The threshold value can be determined by analyzing the image's histogram. A histogram is a graphical representation of the intensity distribution in a digital image. The normal way of obtaining the threshold value is to use the peak and valley method, where the two highest peaks are found and taking the intensity value of the deepest valley between the two peeks. Figure 2.10 shows an example of using this method on an image to get a segmented image.

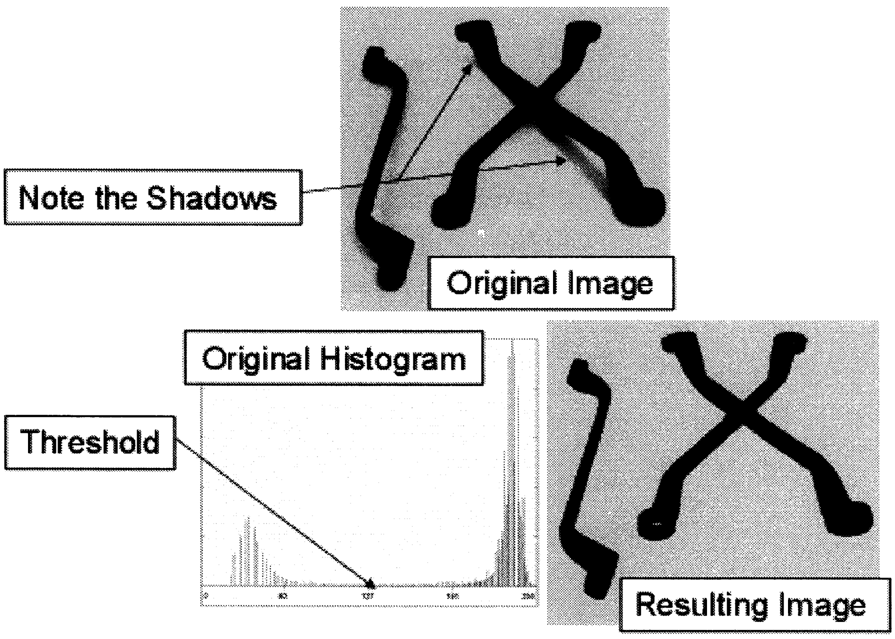


Figure 2.10: Example of thresholding

Thresholding works very well when the object and background has distinctive intensities, and if the correct threshold value is chosen by an algorithm or the user. It does not work so well when objects are connected together.

2.4.3 Morphological Processing

Mathematical morphology, which uses set theory, is used as a tool in image processing to extract components that are useful in the description and representation of region shapes, such as boundaries and skeleton. Some morphological processes used in this project will be discussed. It is assumed that readers are familiar with the basic concept of set theory.

2.4.3.1 Dilation

The dilation of A by B is defined as

$$A \oplus B = \{z \mid [(\hat{B})_z \cap A] \subseteq A\} \qquad (2.4\text{-}1)$$

The dilation of A by B is the set of all displacement, z, such that the reflection of B overlap by at least one element. B is commonly referred to as the structuring element. Figure 2.11 shows a simple set A, the structuring element B and the result of dilating A by B. Dilation is basically the expansion of set A.

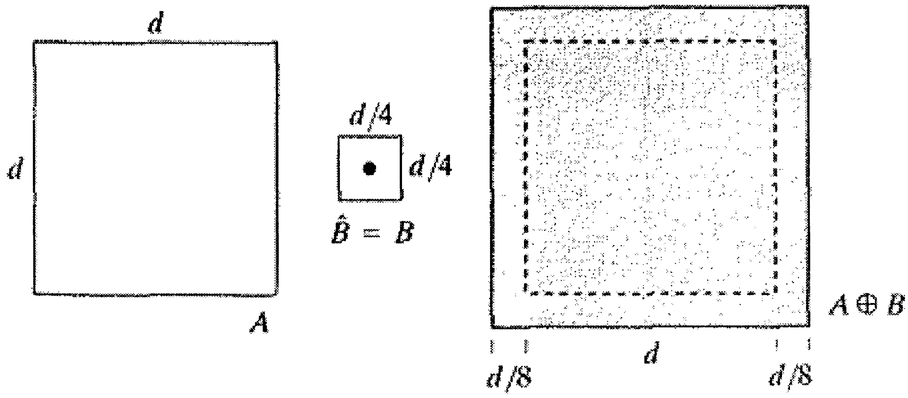


Figure 2.11: Dilation of set A by a square structuring element B

2.4.3.2 Erosion