INTERPRETATION OF TACTILE DATA FROM AN FSR PRESSURE PAD TRANSDUCER USING IMAGE PROCESSING TECHNIQUES

by

Apurva Mahendra Desai

Bachelor of Engineering (Electronics Engineering)
University of Bombay, India, 1992

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE
in the School
of
Engineering Science

SIMON FRASER UNIVERSITY
November, 1994

APPROVAL

Name: Apurva Mahendra Desai

Degree: Master of Applied Science

Title of thesis: Interpretation of Tactile Data From an FSR Pressure Pad Transducer Using Image Processing Techniques

Examining Committee: Dr. William Gruver
Professor, Engineering Science
Graduate Chair, Chairman

Dr.Shahram Payandeh
Assistant Professor, Engineering Science, SFU
Senior Supervisor

Dr. Jacques Vaisey
Assistant Professor, Engineering Science, SFU
Senior Supervisor

Dr. Glenn Chapman
Assistant Professor, Engineering Science, SFU
Internal Supervisor

Dr. M. Parameswaran
Assistant Professor, Engineering Science, SFU
Examiner

Date Approved: Nov. 10, 1994

PARTIAL COPYRIGHT LICENSE

Title of Thesis/Project/Extended Essay

"Interpretation of Tactile Data From an FSR Pressure Pad Transducer Using Image Processing Techniques"

Author:

_____
(signature)

Apurva Desai
(name)

November 7, 1994
(date)

Abstract

The thesis investigates the use of an array of force sensing resistors (FSRs) as a tactile imaging device. A complete imaging system using the FSR array has been developed as a combination of hardware and software. The data acquisition circuit, implemented in hardware, uses a combination of analog and digital circuitry to scan each element of the array. The raw data is then processed using software techniques making use of image processing algorithms.

Objects with simple circular and rectangular footprints are used as test objects. Algorithms have been devised to locate and identify the objects, placed within the array, by finding its dimensions and the position of its centroids. Two methods have been developed to locate the objects, namely, a heuristic method and a model based method. The heuristic method uses a priori knowledge about the object footprint, which is incorporated using morphological filtering. In some cases, subpixel image processing techniques are used to artificially increase the size of the array due to the large dimensions of each element and the small number of total elements.

The results obtained using the heuristic approach are found to be fairly accurate. The centroids and the radii measurement are found to be accurate within the size of one element, namely 1.3cm. The orientation values for rectangular objects are found to be accurate to 10° whereas the dimensions are within 3.8cm, but are found to be fairly close to the recorded footprint.

An object identification system is developed based on features like the weight and the area of the objects. A database that stores the first and the second order statistics of the object features is created and the unknown object is identified using the minimum distance classifier. The results suggest that the system can be used to identify objects.

The second method to locate and identify objects uses Bayesian image restoration

techniques. The images are modeled as a Markov Random Field (MRF) with the probability distribution governed by the Gibbs distributions. The discontinuities at the edges in the images are modeled using the line fields. No other preprocessing is carried out on the raw image. The results on real data suggest that this technique can also be exploited to locate and identify the objects.

Dedication

For my mother ...

Acknowledgments

I am forever indebted to my mother, father, brother and sister in law without whose emotional support, this thesis would not have been possible.

I would like to thank my supervisors, Dr. Jacques Vaisey and Dr. Shahram Payandeh, who put up with a lot and provided constant guidance throughout this thesis. Thanks to my friends, Dave Hargraeves and Graham Finlayson for their help with some of the technical content in this thesis. Thanks to Louis Brassard for some excellent suggestions and the time he spent to discuss my thesis with me. I am also grateful to my friend Aamir Husain for spending countless number of hours trying to debug my code. My sincere thanks to Dr. Ash Parameswaran for keeping my spirits up at times when I was feeling down.

Finally, this thesis would not have been complete without Brad Delanghe and Scott Evenson who ensured that I had a life outside of school.

Let's Rock ...

-Al Bundy

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Robots have become an important part of industrial automation; as a result of which, the field of robotics research has gained a lot of momentum in the last two or three decades. These advances have aided in changing the concept of a robot radically. A robot today is no longer looked upon as a machine simply preprogrammed to carry out menial tasks; instead, it is being looked upon as a human like machine, capable of making decisions and learning with time. Human beings have five senses which are responsible for making most of their decisions and learning capabilities; namely; the senses of touch, sight, smell, sound and taste. Hence, to make robots more human-like, it is important to artificially develop these senses in them. The sense of sight or the human eye is arguably the most dominant of all human senses and a lot of our decisions are simply based on a preliminary observation of the surroundings. It is for this reason that the research carried out in robot sensing has put a lot of emphasis on developing a prototype robot vision system. However, for a robot, an ability to manipulate different objects can not be underrated. A robot has to "feel" the object before it can manipulate it. For us humans, this task is performed remarkably well by skin and hands. In comparison, a look at the development of sensor(s) and sensing system(s) for a robot hand or more generally, the field of tactile or touch sensing, as of today, lacks a quality sensor and finds very restricted usage in industry.

1.1 Motivation

Many different definitions of tactile sensing are commonly used. According to Harmon (1982) tactile sensing is "the continuous-variable sensing of forces in an array", while Russell (1990, page2) defines it in a more general fashion as: " Tactile sensing covers any sensing modality which requires physical contact between the sensor and an external object".

A good tactile sensing system should provide as much information about the object and the surrounding as possible. To completely describe the object under consideration, attributes like weight, color, temperature, thermal characteristics, compliance, resilience, the coefficient of friction of the object surface, electrical conductivity and surface texture need to be known. In the context of robotics, properties like the location, orientation, size, weight distribution and surface shape should also be determined. The latter properties are often determined by using vision or touch. However, information like the forces acting between the robot gripper and the object are best determined by tactile sensing(Russell 1990, page2-3).

The tactile sensors developed so far all differ mainly in the type of transduction technique employed. The input domain is either mechanical, thermal, magnetic, chemical or even radiant, and the output is normally electrical. A vast area of tactile sensing is devoted to novel and practical techniques for object identification. Manipulation of the object is relatively easy once the object, its orientation, exact location and the weight are known fairly accurately.

The above discussions leads to the question regarding the requirements of an ideal tactile sensor. The tasks mentioned above represent a wide variety, varying from object manipulation to extracting features to determining the surface properties and friction. The following presents the result of a survey conducted by Harmon (1982) for a list of desirable characteristics in a tactile sensor. The sensor should have:

- A spatial resolution of $1-2$mm, which is approximately equal to that of a human figertip;
- An array size between 5×10 and 10×20 points per fingertip compared to human fingertip which contains approximately 10×15 touch sensitive points;
- A threshold sensitivity of about $0.5-10$gm per force sensing element;

- A dynamic range of 1000 : 1;
- A stable, monotonic and repeatable sensor response with no hysteresis;
- A sampling rate between 100Hz and 1KHz;
- Broadly, the sensor should be skin-like, rugged and inexpensive.

Besides all of the above, the the cost of manufacturing one such sensor plays also an important factor. In general, a sensor that attempts to incorporate most of the above characteristics tends to be quite expensive and application specific. It is largely due to such stringent constraints that the field of tactile sensors is underdeveloped.

The development of a sensor/sensing system that attempts to meet these requirements at a low cost is an important goal and this is the main motivation behind the work done in this thesis. We have developed a sensing system based on an array of force sensing resistors (FSRs) as the sensor. In the subsequent chapters, we evaluate the performance of this system and suggest techniques for object identification and localization.

The following section presents a brief overview of the existing tactile sensing systems. It focuses on various technique of transduction, associated systems and some of the salient features.

1.2 Background

The background section is divided into the 3 different parts. In the first section, we explain various techniques that have been employed for tactile sensing purposes and point out the merits and demerits of each technique. The next section explains some of the systems and their typical applications. Finally, we present a discussion on the work done with the force sensing resistor used in the thesis and present some concluding remarks.

1.2.1 Transduction Techniques

Some of the commonly used transduction techniques employed for tactile sensing are capacitive, magnetic transduction methods, magnetoelastic and magnetoresistive, mechanical, optical, resistive and conductive, piezoelectric and pyroelectric effects.

What follows are some examples of each kind of transduction technique and the advantages and disadvantages of using them.

Capacitive Transducers

The transducers under this category are concerned with measuring the sensor capacitance, which is made to vary in some fashion with the applied load. An 8×8 sensor has been developed at Bell laboratories by Boie (1984). The contact surface of the transducer consists of an elastic outer layer which is mounted on top of a dielectric layer. The array is formed by orthogonal conductive strips on each side of the dielectric layer. Force readings are measured by calculating the variations in the elastic dielectric layer with capacitance being proportional to the mechanical displacement of the dielectric layer. The size of each element is $2.5mm^2$ and it can measure upto $5kPa$. Another design suggested by Jayawant and Onori (1986) of capacitive transducers include the use of moving dielectric which is spring mounted. Each element is associated with two conductive plates between which the dielectric is mounted. The dielectric moves up and down with the applied force. A 4×4 prototype sensor was built on a large area with some suggestions for a future model with small areas and larger resolution.

The main advantages and disadvantages of the capacitive techniques are presented as follows (Nicholls and Lee 1989):

- Advantages
  - Wide dynamic range, linear response, Robust
- Disadvantages
  - High susceptibility to noise, Dielectrics are often temperature dependent, Capacitance is a function of physical size and hence the spatial resolution.

Magnetic Transducers

The category of magnetic transduction is subdivided into two main types. In the first type, the magnetic flux associated with the sensor is varied by the movements of some mechanical part of the sensor whereas in the second type, use of specialized magnetic materials is made. Under the influence of load, these materials undergo some change

in their magnetic properties. They are termed as mechanical displacement and magnetoelastic and magnetoresistive types respectively (Nicholls and Lee 1989). One of the earlier sensors based on mechanical displacement is described in Sato and Pugh (1986). It is an 8×8 matrix of probes, each driven by a coil. The probes are moved perpendicularly when under the influence of force to record the contour of the object. A voltage proportional to the displacement is induced and is measured by sense coil. It is a rather cumbersome sensor and marks one of the first designs of its kind. Magnetoelastic sensors either exploit the hall effect else use a simple induction technique as explained by Checinsky and Agrawal (1986). A transformer like device was made using nickel-based core and the voltage induced in the secondary coil was affected by the direction of the vector of magnetic induction. A single element prototype was built and had no moving parts. A typical design is presented by Hackwood, Beni, and Nelson (1986) where a tactile array made up of magnetic dipoles embedded in an elastic medium, whose position and orientation are detected by magnetoresistive sensors mounted on the same substrate as that of the dipoles. The dipoles are displaced under the influence of external force and the change in the form of shear, torque and normal pressures can be detected by the magnetoresistive materials. A 7×7 prototype array is developed where the size of each element can be made as small as 0.5mm. A low noise sensor has been proposed by Clark (1990) which uses split-drain MOSFET sensor. 32×32 and 64×64 have been implemented using a $3\mu m$ technology. Scanning times in excess of kHz have been implemented.

The main merits and demerits of either type of magnetic transducers is listed as follows (Nicholls and Lee 1989):

- Mechanical Displacement
  - Advantages
    - Wide dynamic range, Large displacements possible, Simple design
  - Disadvantages
    - Poor spatial resolution, Mechanical problems with the sensor
- Magnetoelastic and Magnetoresistive
  - Advantages
    - Wide dynamic range and low hysteresis, Linear response and robustness, Normal force, shear and torque measuring capabilities

- Disadvantages
  - High susceptibility to external noise and stray fields, Need AC circuitry.

Mechanical Transducers

Simplicity is the biggest advantage of these sensors. The simplest of these sensors is a spring loaded switch that gives an "ON/OFF" reading for contacts. The tension of the spring can be varied to change the sensitivity of the device. The use of mechanical transducers is mainly restricted for simple applications. The advantages and disadvantages are presented below (Nicholls and Lee 1989):

- Advantages
  - Simple and well known technology
- Disadvantages
  - Complex for array applications, Limited spatial resolution and mechanical problems

Optical Transducers

One of the few optical transducers that is commercially available is developed by Rebman and Morris (1986). There are two of these products available. The first type of sensor uses electromechanical transduction. The sensor surface is made up of a compliant material with a grid of elongated portions forming pins. These pins intersect the light path of a photoemitter-receiver pair under the influence of force. The amount of light reaching the receiver is directly proportional to the amount of deflection and hence the force. The second type of sensors are different only in sizes and resolutions. A dedicated processor is available for information processing. Another typical product using optical transduction is described by Begej (1984). The device is made up of a tactile "table" with a membrane above a plate and 128×128 charge injection device camera below to capture the images. The membrane type essentially affects the quality of the recorded image. The advantages and disadvantages of optical transducers are given as follows (Nicholls and Lee 1989):

- Advantages
  - Very high resolution and compatibility with the vision technology, No electrical

interferences, Low cabling requirements

- Disadvantages
  - Dependence on the membrane in some designs, Some hysteresis

Resistive and Conductive Transducers

The field of using resistive and conductive transducers has been heavily studied and some of the designs have been quite successful. The technique involves measuring resistance through or across the thickness of a conductive polymer. Some of the early designs is found in Briot (1979). A development by Robertson and Walkden (1986) shows an array of 256 elements made using carbon fibers. Two sets of carbon fibers are used and are placed orthogonally to each other. The voltage is measured between the two fibers using external electronics. The pressure values vary from 0 to 600kPa and the entire device has been integrated to a local imaging system. Another sensor was developed at the CMU laboratories by Christ and Sanderson (1982) who used carbon impregnated foam pads arranged such that perpendicular conductors are on the opposite ends of the foam. A prototype 16×16 array with a $\frac{1}{4}$in resolution has been manufactured. The sensor worked well for simple object recognition system. Another attempt at the sensor was made by using silicon rubber pads and some such references can be seen in Chalupa, Marik, and Volf (1982) and by Van Brussel and Belien (1986). The first design consisted of 16×32 sensors on the palm of a hand and used pattern recognition principles to identify simple shapes like circular, rectangular, triangular and squares. The second system used a 16×16 elements on a resolution of 1.2mm and a pressure range of 10 to 500kPa. Some of the merits and demerits of using resistive or conductive techniques are described as follows (Nicholls and Lee 1989):

- Advantages
  - Wide dynamic range and durability, Good overload tolerance, Compatibility with IC technology especially VLSI.
- Disadvantages
  - Dependence on the membrane in some designs, Some hysteresis, Limited spatial resolution when compare with vision sensors, Non linear response in most cases.

The FSR array used in the thesis uses a conductive polymer and hence belongs to this category.

Piezoelectric and Pyroelectric transducers

These sensors are often confused with resistive sensors; however they work on a totally different principle. Piezoelectricity is a property by which a voltage is generated proportional to the pressure applied without any needs for external voltage. Pyroelectric is a generation of voltage when an element is heated or cooled. A typical example of a piezoelectric sensor can be found in Dario, Bicchi, and Buttazzo (1986). The device is made similar to the human skin and measures pressures and thermal changes. The advantages and the disadvantages of this techniques can be summarized as follows (Nicholls and Lee 1989)

- Advantages
  - Wide dynamic range and durability, Good overload tolerance, Temperature variations along with pressure measurement capabilities
- Disadvantages
  - Difficult to separate the two effects, namely, piezoelectric from pyroelectric, Inherently dynamic, Complexity in scanning.

Quite often the data obtained from a tactile sensor is corrupted with noise and needs to be processed. From the literature, it is seen that the field of tactile data processing is quite underdeveloped. In the following subsections, we discuss some of the passive processing techniques.

1.2.2 Processing of Tactile Sensory Data

The processing of the tactile data can be done in either actively or passively. Active processing means traversing the surface of the object to build 3-dimensional images and to obtain information about the surface texture similar to haptic exploration. In our thesis, the object is simply placed on a sensor and hence it is categorized under passive sensing. The following paragraphs, therefore, present some of the passive processing techniques that have been used to process tactile sensory data.

Main thrust of research in passive sensing has been on object identification through feature extraction and/or pattern matching algorithms. Briot (1979) worked on a 10×10 resistive sensor to identify one of the 8 faces of an object known a priori to the sensor. The position and location of the object were calculated using the centroid and the second moment about the major axis of the region present in the tactile image. The sensor was presented with each object 10 times in the learning phase. In the recognition phase, the probability associated with each of the features was calculated and the one with the highest possibility was identified. In another paper (Chalupa, Marik, and Volf 1982), a linguistic pattern recognition scheme was employed. The scheme included preprocessing the information by digitizing it and finding the boundary. A Freeman chain coding was employed with each of the pixel in the boundary and then the image was low pass filtered to remove noise associated with digitizing. The image was then high-pass filtered to retain only those changes that contain the boundary of the element. Each boundary thus found was coded using linguistic primitives and a sentence describing each boundary was noted. A paper by Christ and Sanderson (1982) describes the object identification based on feature extraction. A learning phase is formed by representing each object to the sensor 10 times and a statistics of these readings is noted down in a database. The object is then identified based on the closest match with the statistics. In an ongoing project of ESPRIT 278 project that uses both the visual and the tactile information, (Ghani 1988b) & (Ghani 1988a), the tactile image was processed after digitizing and and a pair of finger mounted tactile sensors was used to reduce the features to a typed vectors and matching with geometrical constraints.

Image processing techniques have been quite popular to identify edges and boundaries of the object. In most cases, the processing of the data ends at finding the boundary. One such paper is described by Berger and Khosla (1988) and it presents a variable thresholding algorithm as the first stage. A filtering algorithm that checks for the boundary by considering a simple neighborhood structure.

The array used in this thesis is associated with a lot of noise points and voids and hence require more sophisticated image processing techniques as will be described in latter chapters.

In the following section, we present some of the work that has been done using the FSR sensor and the FSR array that is used in this thesis.

1.2.3 The Force Sensing Resistor (FSR)

The FSR array used in the thesis is a commercially available device manufactured by INTERLINK Inc. There are three types of force sensing devices available. FSR, Linear potentiometer type and the XYZ pads. The XYZ pads and the linear potentiometer type are also classified as the force and the position sensing resistors or simply the FPSRs. For the linear potentiometer type of device, a voltage is applied between the "hot" end and the "ground" end. When a force is applied to the device, a voltage proportional to the distance of the force from the ends is produced. Thus the FPSR device of the linear potentiometer type gives an indication of the force as well as the position along a straight line. The XYZ pad on the other hand, an indication of the position of the force in x as well as the y directions are obtained. Some work done using FSRs can be found in (Speeter 1989) who discusses a piezoelectric type of sensor. The interfacing circuit for the 16×16 array is discussed with possible drawbacks. Some simple experiments are performed to detect an object with a hole at the center and rectangular object. The processing of the data is performed by using a low pass filter. The paper presents a detailed description of the manufacturing of the FSR and little emphasis on the processing the information. It aims to develop a better sensor by improving the conductive film used and suggests a development of an object identification system. The main aim for developing the sensor was to be able to mount it onto the MIT/UTAH dextrous hand.

Some work done using a single element FSR is published by Yaniger (1991). A description of the sensor and its manufacturing process is described. The paper is devoted to evaluating the performance of the FSR and talks about the development of FPSRs. Some work done on an FPSR is presented by Liu and Meusel (1993) and describes some initial results on measurement of the force and position using the FPSR mentioned.

The array used in the thesis presents an attractive option as it is fairly low cost and an easily available device. The thesis describes work done in development of a tactile imaging system using FSR array. Currently, the size of the array is fairly big, and hence imparts a degree of crudeness to the readings. The processing techniques described, attempt to recover the data from such "noisy" measurements. However, the size of the array is expected to be reduced in future and hence an improved set

of initial readings can be expected. This should further enhance the accuracy of the algorithms.

1.3 Contributions

The field of tactile sensing needs a sensor that is commercially available, easy to use and not very expensive. A complete tactile imaging system based on the array of force sensing resistors is described in the thesis. Hardware circuit is developed that reads information off each individual pixels on the array. The algorithms to extract information work with fairly "noisy" measurements but still manage to generate fairly accurate results, albeit, on simple shapes and when the desired information is fairly limited. The ability to locate simple objects using the combination of the hardware and the software is, we feel the most important contribution made by this thesis.

Almost all the existing robotic workcells utilize some prior information about the objects present in the system. However, the morphological filtering methods to incorporate the prior knowledge are used for the first time. The paper by Desai, Payandeh, and Vaisey (1994) shows that the technique has been effectively applied to extract features from noisy measurements. The success of mathematical morphology in the field of object localization is another contribution made by this thesis.

Some work on using Bayesian techniques to restore noisy images has been attempted. The thesis uses a model based approach by treating the array as a Markov random field with probability governed by Gibbs distribution. The incorporation of image discontinuities using the line field model seems to have a lot of promise. Some initial work suggests that the features like the area and the centroids along with the shape of the object can be detected using this technique. This we feel is another important contribution made by the thesis.

1.4 Thesis Outline

Chapter 1 discussed the motivation behind the research and provided the relative advantages of a system having tactile sensing capabilities. It described the background work done in the field of tactile sensing by accounting for the existing

systems. It presented the relative advantages and disadvantages of using different transduction techniques. Finally, the chapter presented some contributions made by this thesis.

Chapter 2 presents an overview of the force sensing technology and presents the hardware issues associated with the system configuration. It discusses the "noise" problems and other potential sources of errors in recorded measurements. Some of the design considerations like the choice of bias resistors is explained in depth. A calibration curve describing the pressure versus the voltage variations is discussed in the end.

Chapter 3 presents a background of all the processing techniques used in this thesis. It presents some of the salient features and selection criterion for each of the technique. The chapter provides the reader with the background necessary to understand the actual application of each of the technique in a later chapter.

Chapter 4 indicates how each of the techniques described in chapter 3 can be applied to process real data. It presents the results on locating circular objects using morphological filtering approaches. It moves onto describe the results obtained for objects with rectangular footprints. The chapter then describes an object identification system based on the previous results.

Chapter 5 begins with some background information on the Bayesian image restoration technique. It then presents some of the simulation results and suggests how this technique can be applied to real data images. Finally, some preliminary results on real images are presented followed by a discussion section.

Chapter 6 presents conclusions and future work which can be built upon the basis provided by this thesis.

Chapter 2

Hardware Description

This chapter presents the reader with an overview of the hardware employed in this thesis. The array used in this thesis comes from a family of commercially available force sensing resistor (FSR) devices (Interlink Inc. 1991). XYZ pads and the linear potentiometer type pads are the other members of this family. All these devices work on the same principle, but are suited for different applications. In the following sections, we present the concept of the FSR technology, describe the sensor used in this thesis and explain some of the associated terminologies. We then describe the data acquisition circuitry developed for the FSR array and discuss some of the associated problems. Finally, a technique used for characterizing the array for a pressure versus the voltage curve is described.

2.1 FSR Technology

2.1.1 FSR construction

The FSR is a thick film polymer device that exhibits a decrease in the resistance with an increase in the force, applied normally to the surface of the polymer film. The construction of a single FSR element is as shown in Fig. 2.1 (Yaniger 1991).

Each FSR element consists of two polymer films; one of the films contains a conductive pattern in the form of interdigitating electrodes whereas the other contains a semiconductor polymer. The fingers of the interdigitating electrodes are typically 0.4mm in width and have approximately the same spacing between them. The two

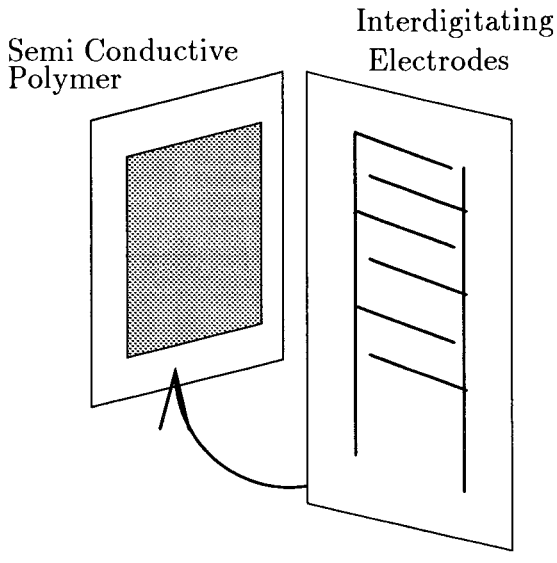


Figure 2.1: Construction of an FSR (Yaniger,1991)

layers are placed on top of each other so that the conductive fingers are shunted by the semiconductive polymer. For zero force, the resistance of the polymer is very high and drops with an approximate power law ($R\alpha\frac{1}{F^2}$) for an increasing force. Fig. 2.2 indicates the variation in the resistance versus the applied normal force for a single FSR element.

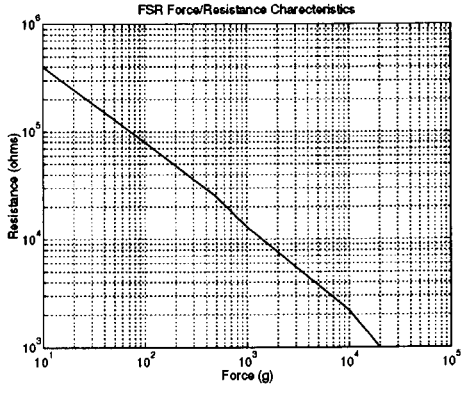


Figure 2.2: Force versus resistance characteristics for an FSR (FSR technical manual)

The curve does not show forces above 10kg of load, since beyond this point the characteristics deviates from the power law and it starts to saturate. The force value at which an FSR element reaches saturation can be changed by varying the spacings

between the interdigitating fingers. The saturation force can be made to vary from 3kg to as high as 50kg.

Table 2.1 shows some of the device specifications provided in the data sheet for the FSR array (Interlink Inc. 1991).

Table 2.1: Some relevant specifications for an FSR

Parameter	Value
Device Thickness	$0.2\text{mm} - 0.8\text{mm}$
Repeatability	$\pm(2-15)\%$
Force Range	$10\text{gm} - 20\text{kg}$
Device Rise Time	$1 - 2\text{msec(mechanical)}$
Lifetime	$10,000,000+$ actuations
Use Temperature	$-30^{o}C to + 170^{o}C$
Maximum Current	$\frac{1\text{mA}}{cm^2}$ of applied force

The array used in this thesis is a 16×16 array of individually addressable FSR elements. A total of 32 lines are provided for interfacing purposes. Each element is $1\text{cm} \times 1\text{cm}$ in dimension and there is a spacing of 0.3cm between each element. The size of the entire array is $20.5\text{cm} \times 20.5\text{cm}$. There is a protective hard plastic coating over the whole array. Fig. 2.3 shows the construction of a portion of the array.

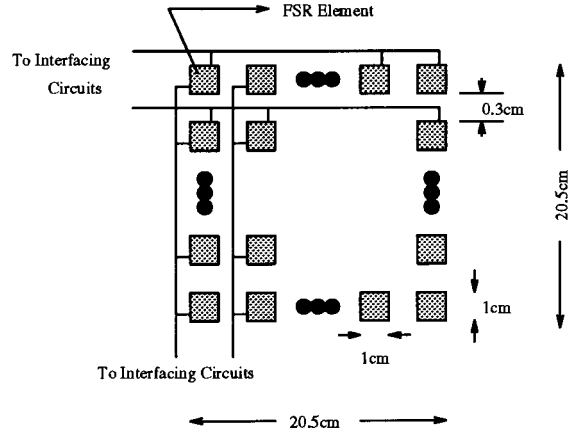


Figure 2.3: The layout of the array

As seen from Fig. 2.3, there is a single line that multiplexes information from all the 16 elements in a row or column.

To read information off each element, some form of data acquisition circuit is necessary. The design of the data acquisition circuit in fact forms the first stage

in processing the information obtained from the array and requires careful choice of various components used in it. It is important to design the first stage of processing with care to ensure a more reliable set of data for all future processing. To understand the operation of an FSR element, it is important to familiarize with some of the FSR terminologies and preliminaries described in the following section.

2.1.2 FSR Preliminaries

Some of the important factors affecting the performance of an FSR and associated terms are as follows.

- Area Effects It is important to realize that the FSR array is approximately a cross between a pressure and a force transducer. The resistance change in the device is roughly inversely proportional to the square of the active area under the sensor, (Interlink Inc. 1991) that is,

$$R_{fsr} \propto \frac{1}{\sqrt{area}} \tag{2.1}$$

  The above equation holds if the force footprint is smaller than the active FSR area but larger compared to the spacing between the fingers. The FSR behaves as a pressure sensor when the applied force acts on a larger area compared to the active FSR area. For all the objects used in this thesis, the latter condition holds true, and as such the behavior of the FSR resembles more to that of a pressure sensor than a true force sensor. The sensitivity of the FSR to the area means that either the FSR must be used as a qualitative sensor, or that by proper mechanical arrangement, the force footprint must be held constant in area, position and footprint. The FSR array is characterized in terms of pressure for the reasons mentioned above.

- Sensitivity indicates the smallest force change that changes the resistance of the device. Sensitivity is a performance characteristic and is indicated by the slope of the voltage versus the pressure curve. It will be explained later how the sensitivity of our data acquisition circuit depends on the choice of bias resistors and the area of the sensor under the object.

- Repeatability is the ability to repeat, within a tolerance, the previous response characteristic. Repeatability provides information about the hysteresis associated with the device. The semiconductive material used has a retentive property which retains some information from the previous load cycle. This means that different voltage values get recorded for the same pressure, one for increasing pressure conditions and one for decreasing pressure conditions. The thesis assumes that the objects placed on the FSR array do not move. In other words, only static conditions are assumed and hence, the hysteresis is not a significant effect. The repeatability in our system is mainly governed by the presence of random "noise" values.

- Actuator/Object Characteristics The compliance of the force actuator (that is, the actual physical component that transfers the force to the FSR, objects in our case) is an important issue while selecting the objects. A rubber sheet or other elastomers are often employed to evenly distribute the force over the entire surface. The effect of such elastomers is more pronounced at lower force values. Later in this chapter, the advantages and disadvantages of employing a foam pad over the entire array is shown[1].

- Conductor Design Fineness of pitch (also known as the "space and trace") of the conductive fingers is a key issue in the design of the FSR elements. For any given area, the finer the the pitch, the greater the number of actuated fingers. Increasing the number of shunted fingers increases the dynamic range of the device; however, the main disadvantage to this is the increase in the cost of the device.

- Coupling Effects This is a problem observed with an array type of sensor device. There are two different types of coupling phenomenon, namely, mechanical and electrical. Different elements of the array are manufactured using the same continuous semiconductor polymer. This means that depressing the semiconductor material under one element causes a depression of the semiconductive rubber under surrounding elements. It was observed that this mechanical coupling contributed little to errors in the readings. The second type of coupling, namely, the

[1] Employing a foam pad over the array is similar to employing a foam pad on the contact surface of the object.

electrical coupling is the chief contributor to erroneous readings. This is observed due to the electrical interfacing circuits employed. As explained in Section 2.3, electrical coupling is dependent on the choice of the bias resistors. Two types of compensating techniques are normally employed to account for the coupling effects; hardware and the software. This thesis employs a software compensation technique. The hardware compensation technique is described under the future work section in Chapter 6.

In the next section, we describe the hardware circuitry employed and describe the function of each of its block individually.

2.2 Circuit Description

The data acquisition circuit for the FSR array involves a software controlled analog and digital hardware circuits. From Fig. 2.3, we can see that a total of 16 pixels are connected to each of the 32 lines. In other words, the information is multiplexed onto each of these lines. To successfully read information off each pixel, it is necessary to scan the entire array sequentially, that is, row by row and column by column. The block diagram of the overall data acquisition setup is as shown in Fig. 2.4.

The circuit is interfaced with a `PC486` and hence, all the voltages are maintained within the standard TTL level. Also, care should be taken to maintain the current through each of the FSR elements to less than $1\frac{\text{mA}}{\text{cm}^2}$, which is specified as the higher end of the current limit in Table 2.1. In the following subsections, we consider each block of the circuit individually, describe their functions and discuss some important design considerations associated with them.

2.2.1 Demultiplexing Circuit

The heart of the demultiplexing circuit shown in Fig. 2.5 is `IC 4067` which is an analog multiplexer/demultiplexer chip. The clock pulse for the IC is software generated and is provided through an I/O port of the `PC486`. The input to the demultiplexer is $+5V$ and is inverted on each of its 16 output lines. The importance of the inverter circuits can be explained as follows:

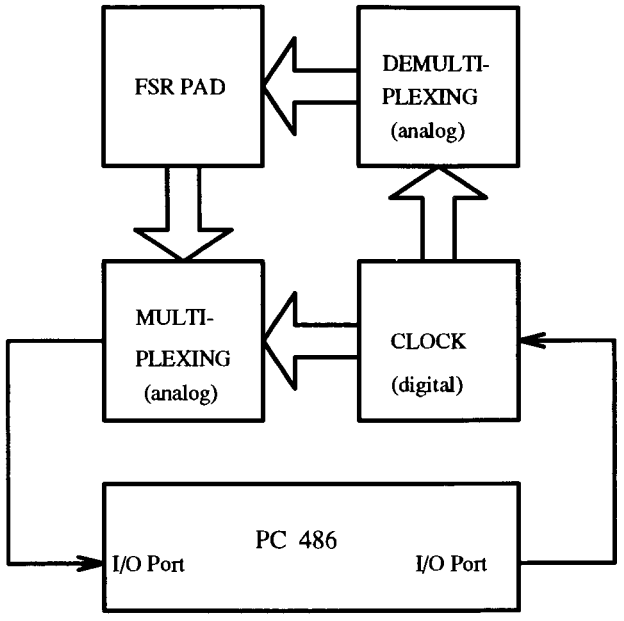


Figure 2.4: Block diagram of the data acquisition circuit

1. It is convenient to see an increase in the output voltage with an increase in the applied pressure instead of the other way around. To achieve this, we should increase the current in the "output" (multiplexing) circuit for an increase in the applied pressure, which is possible only by creating a positive potential difference between the "output lead" (multiplexing end) and the "input lead" (demultiplexing end) of each FSR element. However, a potential of $-5V$ can not be directly applied as the input lead of the FSR element directly since `IC 4067` has a maximum negative voltage rating of $-0.5V$ [2]. An inverter circuit thus becomes necessary to invert the voltage on each line of the demultiplexing circuit before it is applied at the input lead of the FSR element.

2. The second advantage to using an inverter circuit is the buffer action between the FSR element and the demultiplexing circuit. The back current which might flow from the FSR element into the demultiplexing circuit is reduced by employing the inverter thereby reducing the leakage paths and mitigating the effects of a potential source of error.

[2]This is true of most TTL level devices. They are not suited for voltages in excess of $-1V$ in general.

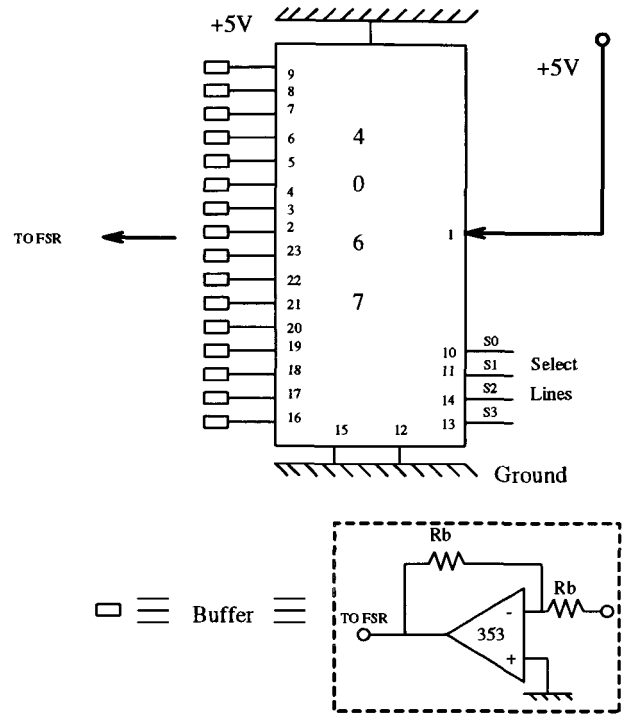


Figure 2.5: Demultiplexing circuit

2.2.2 Multiplexing circuit

The IC 4067, used in multiplexing mode, can selectively output the information from each element onto the input of the A/D converter board on PC486. The multiplexing block of the circuit is shown in Fig. 2.6.

It is necessary to use an analog IC for the operation since the current flowing through the FSR devices for different pressures is an analog value. Each line of the multiplexing circuit employs a current to voltage converter since it directly connects to an input port of an A/D converter card which reads these voltage values. An alternative design would be to use just one current to voltage converter at the output of the multiplexing circuit. Although reducing the amount of hardware employed in the multiplexing section, this causes a lot of leakage current paths and the readings thus become less reliable.

The operations of the multiplexing circuit and demultiplexing circuits are controlled using a software controlled clock circuit.

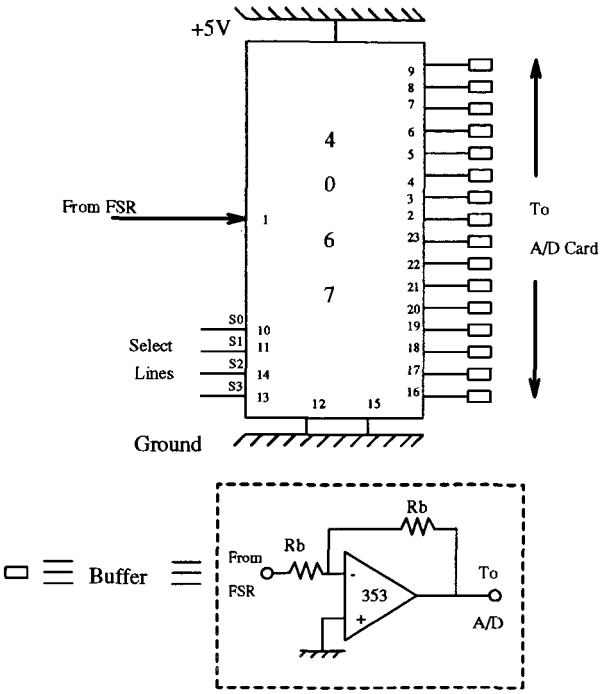


Figure 2.6: Multiplexing circuit

2.2.3 Clock circuit and the Software Block

The operation of the multiplexing and the demultiplexing circuits explained above are controlled using a digital clock `IC 74393` which is in turn controlled by the software written on `PC486`. The clock and the output of the multiplexing circuit are connected using one of the I/O ports of the `PC486`. The software section performs some other functions besides controlling the operation of the circuit. The necessity and the tasks performed by the software block of the circuit can be summarized as follows.

- There is usually a speed mismatch between the A/D card and the rate at which the data is read off the output of the multiplexing circuit. This could possibly result in a lot of ambiguity in the recorded results in absence of some form of handshaking between the two devices. A faster multiplexing circuit could overwrite the previous information transferred to the A/D card before it has been read into the `PC486` whereas a slower multiplexing circuit would mean that A/D card ends up recording the same information for a different element. The software prevents this by controlling the clock inputs to the multiplexing and demultiplexing blocks. For every clock pulse to the demultiplexing block, a total

of 16 (since there are 16 elements per row of the array) clock pulses are applied to the multiplexing circuit. This ensures that the data is read off the A/D card before new data is transferred into the A/D card.

- To minimize the effects of accidental noise peaks during a measurement cycle, it is useful to sample each point, a sufficient number of times, before reading off its average over those samples. The software written averages the reading over 100 samples taken at the same element. The values of one 100 was heuristically chosen and it was noted that in practice, a lesser number of samples, namely 10, worked equally well. The scanning time for the entire array was found to be less that one minute with scope for further improvements in speed. No experiments were performed to optimize this time as the objects in this thesis are considered to be stationary[3].

The choice of resistance (marked R_b in Fig. 2.5 & 2.6) in the inverter and the current to voltage converter circuits is an important design issue. These resistors are termed bias resistors because of their direct effect on the sensitivity. The following section describes the methodology employed in selecting an appropriate bias resistor.

2.3 Choice of the Bias Resistors

The value of the bias resistors is an important design consideration as it directly affects the sensitivity as well as the electrical coupling between the elements. The sensitivity is directly proportional to the value of the bias resistors, that is, higher values of the bias resistors result in higher recorded voltages for the same force. On the other hand, the electrical coupling effect between the elements also increases with an increase in the bias resistor value. Fig. 2.7 indicates the effect of electrical coupling.

Fig. 2.7 illustrates a simplified diagram of an active FSR element. It shows the leakage effect due to elements in the same column. Electrical coupling increases the difference between the actual value and the recorded value, or in other words, the error in the recorded measurement. The choice of these resistors thus involves an engineering compromise between the acceptable sensitivity value and the allowable

[3]Scanning times of as fast as $20Hz$ have been recorded by Speeter (1989) using a similar piezo-electric transducer.

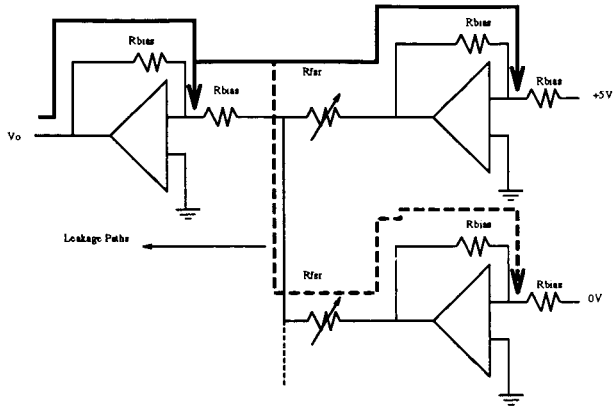


Figure 2.7: Electrical Coupling Phenomena

coupling factor. A computer simulation was performed to arrive at the value of the bias resistors. The remaining part of this section is devoted to various conditions under which the simulation was performed. The graphs following each simulation is an indication of the output voltage variation versus the variations in the FSR resistance for different simulation conditions.

2.3.1 Assumptions

The following assumptions were made while simulating the circuit.

1. It was assumed that the operational amplifiers used are ideal in nature. They were simulated as devices with an infinite gain, infinite input resistance and zero output resistance.
2. The resistors were assumed to be all identical with zero tolerance value.
3. The resistance between adjacent elements was assumed to be infinite and all the lines for interfacing purposes were assumed to be ideal conductors.
4. The resistance for the FSR was varied from a low to a high value for all the simulations. This corresponds to a pressure variation from a high to a low value.

A circuit simulation software called HSPICE was used and the graphs were generated using the graphical interface for the HSPICE simulator. The assumptions made above resulted in a somewhat simplified simulation, but provided a fair idea of the circuit behavior. In the following graphs, the voltage values are plotted on the y-axis

corresponding to various pressure values on the x-axis [4].

2.3.2 Simulations

Four main simulations were performed:

1. Effect on Sensitivity

   The simulation was performed by varying the bias resistance value from 10k to 100kΩ. The lower limit for the bias resistors was chosen to be 10kΩ because values less than 10kΩ resulted in very low values of sensitivity. Value of 100kΩ was chosen as the higher limit because higher resistance values were comparable with the values of FSR resistances under load conditions and resulted in significant loading between the FSR and the bias resistors. It should be noted that the value of FSR resistance drops to as low as 1kΩ for a force of 2kg. The graphs for different bias resistors can be seen in Fig. 2.8. The x-axis represents the variations in the FSR resistance for different loading conditions whereas the y-axis shows the resultant output voltages.

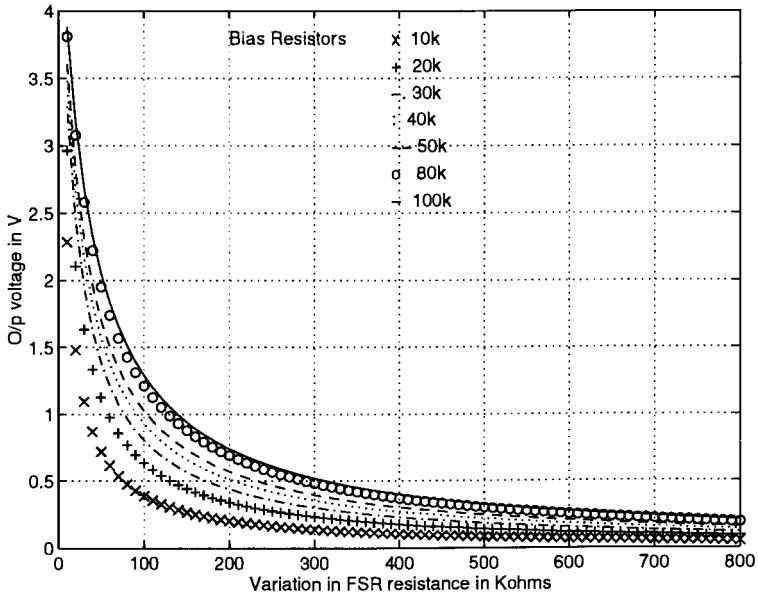


Figure 2.8: Sensitivity for different values of bias resistors

[4]The pressure values are plotted as variations in the FSR values on the x-axis

The results show a direct relationship between the bias resistors and the sensitivity values. For any value of pressure, the voltage recorded is higher for larger resistance value. Of course, the bias resistor value of 100kΩ records maximum sensitivity, whereas 10kΩ records least sensitivity. Also, the sensitivity values seem to change less rapidly for bias resistance values in excess of 40kΩ.

2. Effect on Coupling

Due to the design of the multiplexing and the demultiplexing circuits, electrical coupling was found to exist between the elements of the same column. There was little or no electrical coupling between the elements in the same row or between adjacent columns. The simulation was performed by keeping 8 of the 16 elements of the column under a pressure of 100gm corresponding to a resistance value of 80kΩ. The test element was was simulated for a variation in the bias resistor values. The value of 100gm/element for 8 elements was heuristically chosen as it represents a fairly heavy object. The bias resistance value is varied from 10kΩ to 100kΩ in steps of 10kΩ as before. The graph for the simulation is seen in Fig. 2.9. The graph presented in Fig. 2.8 presents the values without any coupling effect.

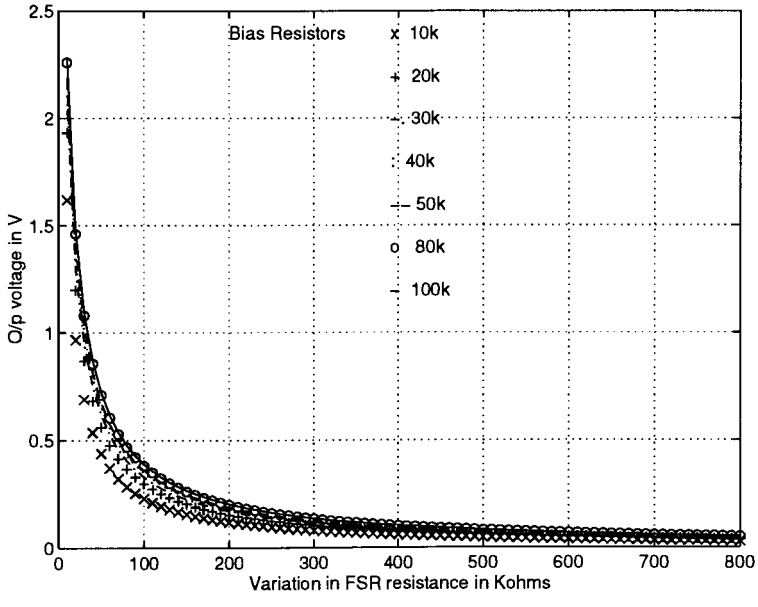


Figure 2.9: Effect of loading on sensitivity

The results suggest that the sensitivity curves are very closely clustered together. In fact, the voltage values are very close to each other for any increase in bias

resistor values beyond 50kΩ. On the other hand, a look at the graph in Fig. 2.8 suggests that the sensitivity increases for an increase in the bias resistance. From Fig. 2.8 and 2.9, we can conclude that for bias resistance values in excess of 50kΩ, there is a more pronounced coupling effect. The effect can be attributed to the smaller currents flowing in the circuit, which results in more errors for small leakage currents. Also, Fig. 2.8 suggests that the sensitivity values for resistances less than 30kΩ are low and hence at this stage the range of interest can be narrowed to between 30kΩ to 50kΩ.

3. Comparison of coupling effects

As mentioned in the last discussion, the range of interest for bias resistor values was narrowed down to values between 30kΩ and 50kΩ. Hence, the next simulation was performed to see the effect of coupling on the bias resistor values between 30kΩ and 50kΩ. The simulation was carried out in steps of 10kΩ or for values of the bias resistances of 20k, 30k, 40k and 50kΩ. The loading resistor for the pixels in the same column was assumed to be the same as before, that is, 80kΩ. The curves for 20k, 30k, 40k, and 50kΩ from Fig. 2.8 and 2.9 are merged together in Fig. 2.10 as shown. The curves corresponding to the higher voltage values for any bias resistance indicate the output voltage values before coupling.

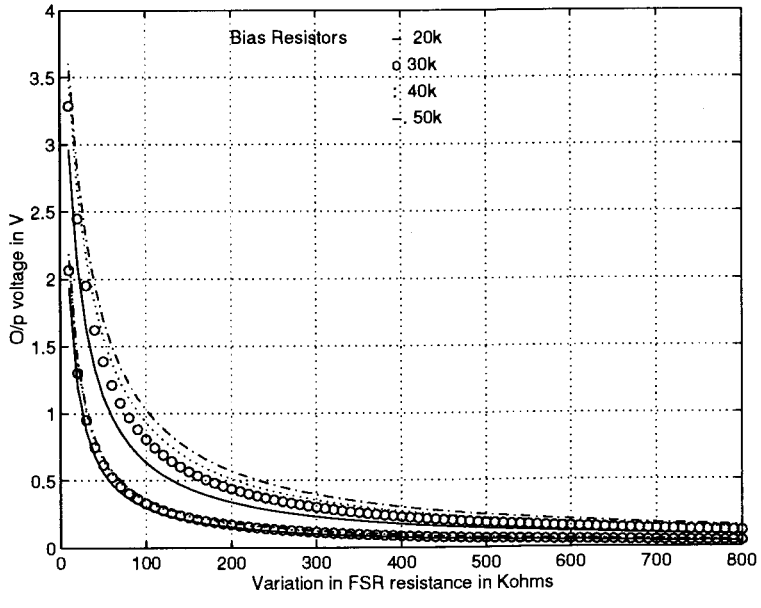


Figure 2.10: Comparison of the curves before and after coupling

The graph indicates that the coupling effect for the bias resistance value of 30kΩ is the lowest amongst all the three. In other words, the difference between the "true" value and the recorded value is the least for 30kΩ. A resistance value in the vicinity of 30kΩ, 33kΩ, was thus selected.

4. Variation in coupling for 33kΩ

The purpose of this simulation was get a qualitative idea of the amount of coupling for a bias resistance of 33kΩ. 8 randomly chosen elements in the same column were subjected to same resistance variations each. The values loading chosen were widely spread out between 60kΩ and 300kΩ. The graph for this simulation is as shown in Fig. 2.11.

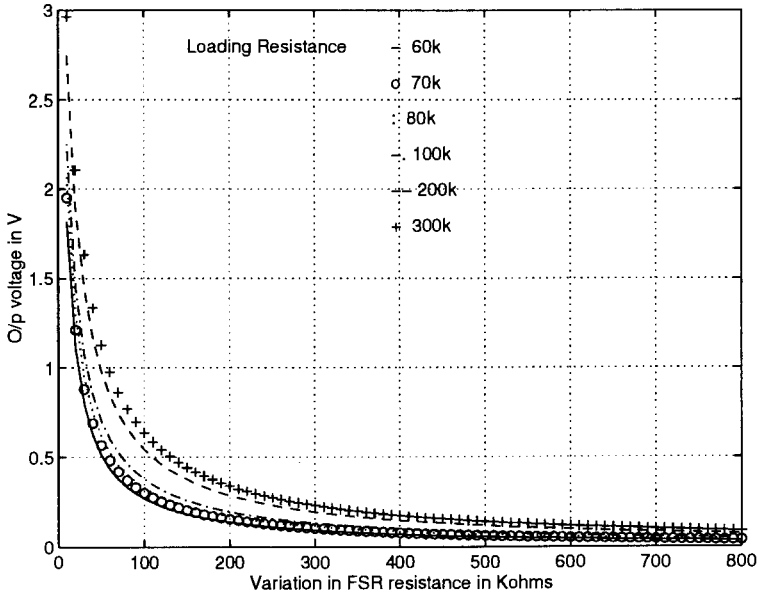


Figure 2.11: Effect of Coupling on the Bias Resistance Value of 33kΩ

The results show more pronounced coupling effect is observed for smaller loading resistance values. But, smaller loading resistance values correspond to larger pressures and thus there is more coupling at higher pressures.

From the above discussions, we can conclude that the choice of the bias resistance is indeed a compromise between the sensitivity and the coupling between the elements. Higher values of bias resistance result in higher sensitivities but increase the coupling effects at the same time. On the other hand, lower values of the resistances result in lower coupling but lower sensitivities. The range of bias values can be narrowed down

to between 30kΩ and 50kΩ as a compromise between the coupling and the sensitivity with the optimum value being application specific. In our case, the value of 33kΩ has been chosen as a compromise between the coupling and the sensitivity. All the future readings are measured with the bias resistance value of 33kΩ unless otherwise specified.

The next important step is to characterize the array. The characteristics is given in terms of the voltage variations for different values of pressures acting on the array. The next section presents the description of our technique used for characterizing the array and discusses some of the associated problems.

2.4 Characterizing the array

In the following sections, we present some of the problems associated with a typical measurement and discuss some of the attempted solutions. We then describe the technique that seemed to produce the "best" results and show the graph of the pressure versus the voltage curve.

The FSR pad used in the thesis is covered with a plastic coating for protection from abrasions and scratches. The size of each FSR element is 1cm × 1cm with a spacing of 0.3cm between any two elements. A typical reading for an object, (weight=1860gm and Radius=6.9cm), with circular footprint, is as shown in the Fig. 2.12.

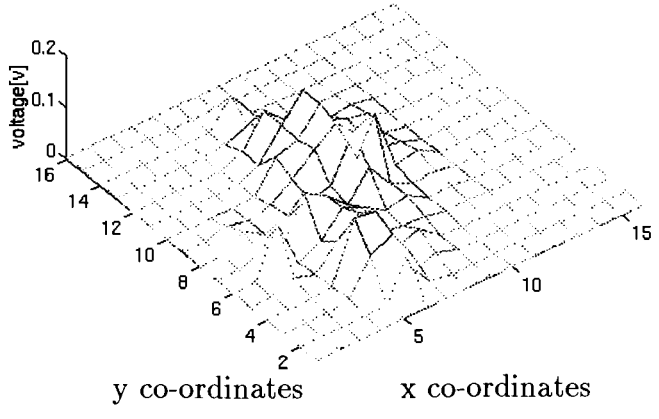


Figure 2.12: A typical image measurement

The image shows that the footprint of the disc does not look very circular and seems to have pressure concentrated over a few points. In fact, it was noticed that

the protective plastic coating used over the array developed minor bumps randomly over the surface. The result of these bumps was the concentration of pressure at some elements while completely depriving some of the other elements from making any contact with the object. It is our hypothesis that the occurrence of "bumps" and "valleys" phenomenon could be due to either or all of the following: uneven surface on the array, presence of dirt and the uneven contact surface of the object. A simple experiment was performed in order to support this claim. One of the pixels on the array was raised slightly using a grain of sand to simulate a "bump". The result of such a bump can be seen in Fig. 2.13.

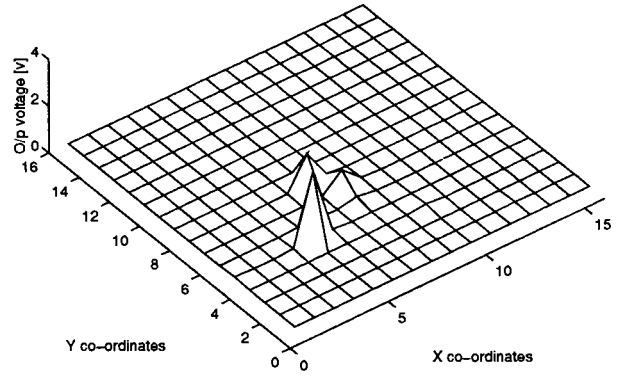


Figure 2.13: Effect of a "bump" on the array

The original image of a disc (weight=2000gm,rad=2.5cm), seems to be completely concentrated on the bump at pixel position(5,5) on the array. The first attempt to mitigate the problem of uneven distribution was sought by using the foam pad as described below. Fig. 2.14 shows the same footprint as in Fig. 2.12, after the use of the foam pad.

From Fig. 2.14 and Fig. 2.12, it is seen that the distribution of the force seems to be more even in case of the foam pad than the one without it. Also, the image appears a lot smoother in the latter case. The smoothness of the image is in fact the biggest disadvantage while using the foam pad. The equivalence of a foam pad in signal processing terms is a simple low pass filter. The low pass filter works by averaging out the values of elements over a window of a certain size. Hence, the image appears to be a lot more uniform. Also, accidental peaks or the "peaks" and "valleys" phenomenon

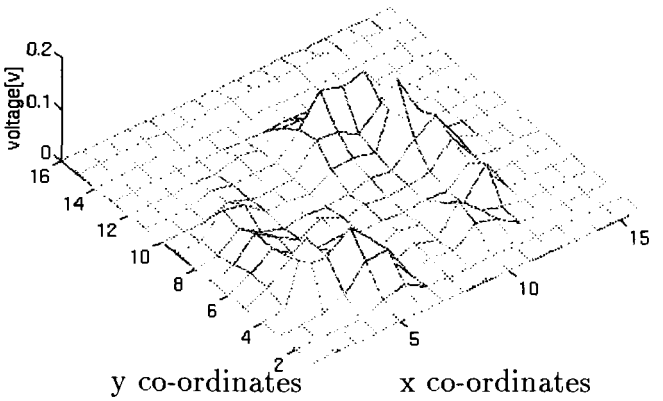


Figure 2.14: Effect of the foam pad

boundaries because of averaging. In other words, the difference between the object and the background becomes less distinctive. This is quite a drawback for non-circular objects. A better and a more general solution is thus needed for a wider set of objects. The following subsection explains the use of cardboard pads over each element.

2.4.1 Use of Cardboard Pads

It is important to recap some relevant details at this stage. It was mentioned before that the FSR is a cross between the force and the pressure transducer. It behaves as a pressure transducer when the object footprint is bigger than the active FSR area. All the objects in this thesis are much bigger in size than each FSR element and hence the FSR array behaves as a pressure transducer. Going back to the basic definition of pressure as given by Eq. 2.2.

$$P = \frac{F}{A} \tag{2.2}$$

where P = pressure, F = Force and A = Area, it is seen that the pressure is dependent on the area of the object. Continuing with the recap, each element of the array is only 1cm $\times$ 1cm and there is a spacing of 0.3cm between the two elements. This gaps is completely devoid of any sensing capabilities. Any contact of the object over this region means a loss of information. The weight estimate of the object becomes erroneous if this loss is not accounted for. A direct software compensation of such information loss becomes quite difficult when the effect is coupled with the presence

of random noise in the form of the "peak and valley" phenomenon. Hence, in order to reduce the loss of pressure information, it is important to physically prevent the object from making any contact with the insensitive region and allow contact only over the surface with the sensing capabilities. This is easily achieved by raising the sensitive region of the pad slightly over the insensitive regions. The simplest way to achieve this is by placing small pieces of cardboard materials over the each FSR element and leave the remaining portion of the array untouched. The choice of the cardboard pads was (heuristically) made, since this material is firm enough not to bend significantly under the application of pressure and cause additional compliance errors, while at the same time it faithfully transfers the pressure information onto the FSR elements.

The next subsection discusses some of the problems faced while calibrating the array with the cardboard pads and the solution to those problems. It concludes with a pressure versus voltage curve that is used in the rest of the thesis.

2.4.2 Calibrating The Array

The first attempt at calibrating the array was made by loading one FSR element with varying weights. It is important to note that measured voltage is very sensitive to the exact position of the object on the sensor. However, maintaining the same position on the FSR element every time it is loaded is a very difficult task. Also, since the area of each element is small, it is difficult to find larger weights with area small enough to lie completely on only one element. This means that we have to stack up weights on top of weights to simulate a heavier object. This however was again a difficult task since, the FSR element is very sensitive to the way its loaded. Slight shifts in the effective center of pressure of the stacked weights resulted in readings that varied by as much as a factor of 2. Thus, it was difficult to make the FSR repeat the same characteristics. The obvious solution is to increase the effective area over which the object lies. This can be achieved if a groups of four neighboring elements are loaded at the same time. The reading can then be averaged over the four elements. This test was repeated a number of times over different groups of elements. A set of 9 curves obtained by selecting a set of 4 elements at random positions on the array can be seen in the Fig. 2.15. The curves are shown for two different weight ranges. The left hand

curve is for a 0 to 700gm of weight whereas the right hand figure indicates a weight variation from 750 to 3000gm.

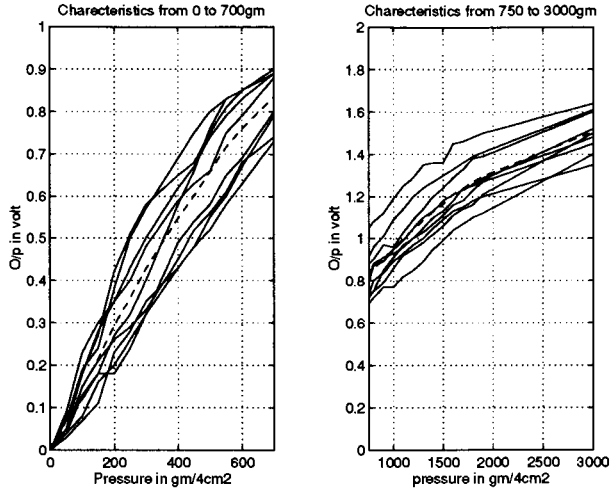


Figure 2.15: Average pressure reading over 4 elements

Characteristic for each individual element can be easily derived from Fig. 2.15 by simply dividing the pressure. The graph for individual element characteristic looks as shown in Fig. 2.16.

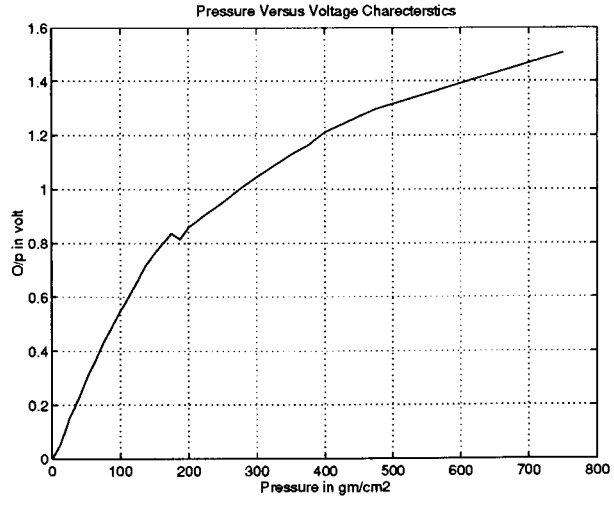


Figure 2.16: Characteristic of an FSR element

A typical curve showing the errorbars is shown in Fig. 2.17

The curve indicates an error bar of $\pm 20\%$ which a dramatic improvement over the $\pm 100\%$ variations observed for a single element characteristics. The curve shows a

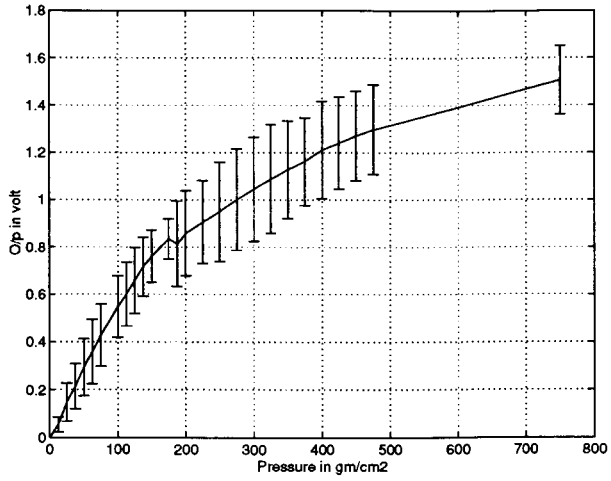


Figure 2.17: Typical error bars for the pressure versus voltage curve

slight glitch at the center which smoothed by fitting a fifth order polynomial to the dataset as shown in Fig. 2.18. The curve shown in Fig. 2.18 is used elsewhere in the thesis.

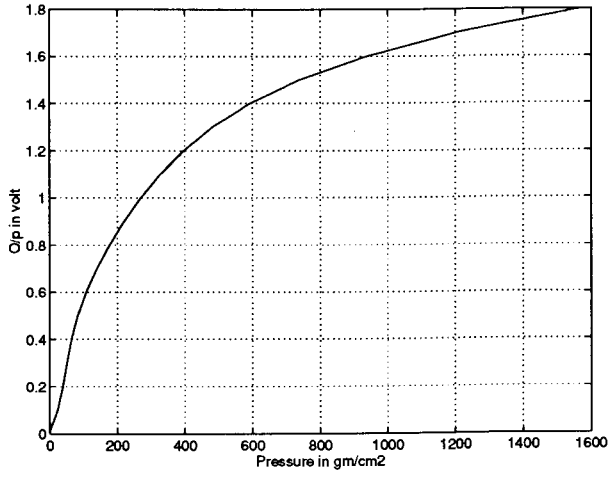


Figure 2.18: Final calibration curve for an individual FSR element

The figure shows the calibration curve that will be used for all future purposes unless otherwise mentioned.

2.4.3 Behavior of the Array for Different Weights and Areas

The calibration curve discussed above was found to behave in a particular fashion when objects of different weights and areas were put on it. The weights of the objects were found to be closer to the actual weight in some cases and quite off in some other cases. In all the cases, the weight was calculated by substituting the voltage values obtained at various elements of the array to the corresponding weight values shown in Fig. 2.18. The weight calculated was dependent on two distinct parameters, namely, the actual weight of the objects and the area of the objects. It is interesting to note that in many cases, the weight readings changed when two objects placed on top of each other were changed into the other position by replacing the top one with the bottom object and vice versa.

To arrive at a sensible conclusion, some basis of comparison between any two objects was needed. For example, consider object A and B of weight $W_a$gm and $W_b$gm and area $A_a cm^2$ and $A_b cm^2$ respectively. In order to compare the effect of different weights and areas on the array, it is necessary to observe what happens at every pixel of the array. Hence, it is necessary to see how much pressure is theoretically developed at each pixel due to each of the objects A and B. The pressures developed due to A and B are given by $\frac{W_a}{A_a}\frac{gm}{cm^2}$ and $\frac{W_b}{A_b}\frac{gm}{cm^2}$ respectively. This provides some basis for comparison between different objects. Five test objects, all with circular footprints, were chosen; whose weight/area are given in Table 2.2

Table 2.2: Test objects for calibrating the behavior of the array for different weights and areas

Object	weight(gm)	Radius(cm)
Object 1	1860	6.9
Object 2	1090	6.1
Object 3	675	4.3
Object 4	1280	3.8
Object 5	1000	2.5

Experiment

The following experiment was performed to evaluate the performance of different weights applied on different areas. Each object mentioned in Table 2.2, was subjected

to heavier weights by piling up other weights on top of it and calculating the weight. A curve is then plotted for actual weights versus the obtained weights for all the objects mentioned in Table 2.2. The curve indicates which object reads closest to the actual weight for any given area. In other words, the effects of varying the areas can be seen from the curve. The curve is plotted in Fig. 2.19.

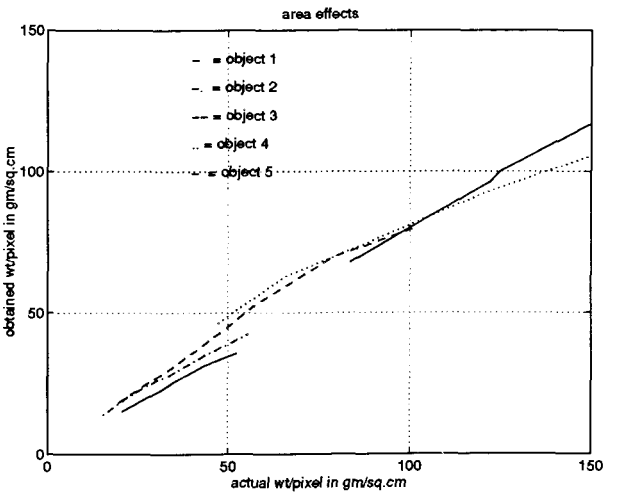


Figure 2.19: Effects of different areas on weight measurements

The curve shows that the weight measurements get closer to the actual weight as the area goes on decreasing. For example, the weight of 20gm per pixel will be more accurate than 40gm spread over two pixels. The curve suggests a decrease in the accuracy of the weight determined when it is spread over a larger area. The effect can be contributed to electrical coupling. The greater the number of pixels covered, the higher the coupling effect and poorer the weight estimate. The conclusion is that smaller objects provide more accurate weight estimates than the larger objects for the same weight per pixel.

In all of the following chapters, the voltage values at every point on the array is converted into intensity values such that brighter points represent higher voltages and lighter points represent lower voltage values. The norm is only for convenience and has no bearing on the output of the processing techniques.

2.5 Summary

This chapter has presented a review of the FSR technology. Some of the relevant features were discussed and the Table 2.1 presented typical values for each parameter. The data acquisition circuitry associated with the setup was described. The circuit was designed and developed to interface the FSR array to a PC486 through an A/D converter card. The software was developed to control the operation of the circuit. It was mentioned that the objects used in the thesis are stationary and hence the scanning time for the entire array is not very important. Some of the design issues related to various blocks of the data acquisition circuit as described in Fig. 2.4 were described in detail.

The next section was devoted to choosing an optimal value of the bias resistance for the circuit. It was shown through HSPICE simulations that the bias resistance value of 33kΩ was the best suited for our setup.

The final section was devoted to developing the calibration curve for the array. Some of the problems associated with each measurement and some of the techniques employed to mitigate their effects were presented. Finally, the pressure versus the voltage curve are presented. The curve was developed for different weights and different areas and it was shown that the weights spread over smaller areas read more accurately than larger weights.

The data obtained from the array is stored in files and processed using image processing techniques. The next chapter describes the background material associated with each of the image processing technique used in the thesis. It presents a general overview of the technique and comment on some of the salient features of each technique.

Chapter 3

Background of the Processing Techniques

This chapter presents a theoretical background for the various techniques employed in this thesis to interpret information obtained from the sensor array. It was stated in Chapter 2 that the voltage value at an element is directly proportional to the pressure over that element. However, the measurement also gives the geometric position in terms of the x and the y co-ordinates with respect to some fixed reference point on the array. Hence, it can be said that each element of the array measures 3-dimensional information about pressure.

The following sections provide some background information on the various processing techniques employed to interpret the tactile images. The actual use and implementation of these techniques is explained in Chapter 4 in this thesis.

3.1 Processing Techniques

The choice of an optimal set of processing techniques depends on the type of application and the expected outcome. Thresholding, morphological filtering, edge detection, convex hull, least squares fit and low pass filtering are some of the techniques used throughout this thesis. Some theoretical background on these techniques is provided below. The actual implementation and the specific usage of these techniques will be described in Chapter 4.

3.1.1 Thresholding

Thresholding is the most commonly used of all image processing techniques and quite often is the first processing block in many such systems. The success of a thresholding operation is largely dependent on the choice of the *characteristic function*, th(x,y). To understand the meaning of a characteristic function, consider a simple image as shown in Fig. 3.1 consisting of just one object (region marked as A Fig. 3.1) with the remaining region (region marked as B in Fig. 3.1) in the "field of view" treated as background.

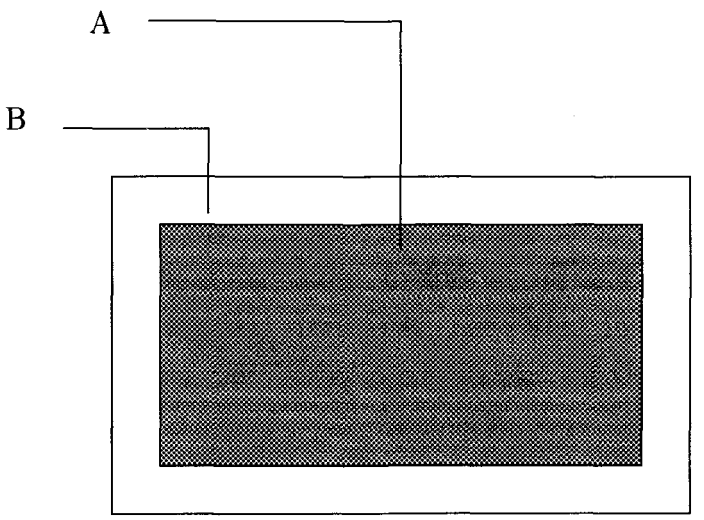


Figure 3.1: A simple binary image with just one object

In such a case, it is easy to come up with a function, th(x,y), that is 0 for all points belonging to the background and is 1 for all the points belonging to the object. This function is called as the characteristic function and the operation of creating a *binary image* using a function similar to th(x,y) is termed as thresholding (Horn 1986, page 47). In the above case, the thresholding operation simply defines the characteristic function to be a 1 wherever it encounters an intensity value of greater (or lower) than a certain fixed value and assigns a value of zero for all values lower (or higher) than that fixed value. The importance of thresholding operation in image world is that it eliminates points that are deemed to be in the same class as the background and separates the values corresponding to the foreground. Thus, thresholding acts like a preliminary filter that suppresses unwanted points from the desired points in the

image.

A typical image processing algorithm is carried out in blocks with the output of one block being treated as the input to the other. It is quite advantageous to employ a simple thresholding filter as the first block of processing algorithm because it helps in reducing the computational burden on all the subsequent points by eliminating the points that it considers not useful. All the subsequent blocks make use of the image generated at the output of the thresholding block. The performance of thresholding operation, thus, has a great influence on the overall quality of the processed image. A poorly thresholded image results could either result in a drastic loss of useful information or increase ambiguity between the unwanted and the wanted points. The selection of the characteristic function, th(x,y) is therefore requires careful consideration. A technique commonly employed is the selection of an automatic threshold value (Horn 1986, page 90) capable of handling images of varying intensity levels and surface illumination properties. In case of digitized images, which is the case in this thesis, automatic thresholding can be performed using a *gray-level histogram* or a *cumulative gray level histogram* (Horn 1986, page 91). In this method, a histogram is formed that gives the number of pixels in each of the gray levels or the "bins". If the image is simple enough, a clear distinction between two separate peaks in the histogram, one corresponding to the object and the other corresponding to the background, is clearly evident. The choice of a threshold value for the characteristic function thus becomes easy and any values between the two peaks can be chosen. An important consideration while developing a histogram is deciding the size of each bin because it directly affects the resolution in the resultant image. Wider bins result in an image with lowered resolution and vice versa. Different techniques for determining the threshold values are normally employed where the intensity levels between the object and the background are not very distinct or in other words, the two peaks in the histogram are not quite distinguishable. The methods usually sought under these conditions are quite application specific.

As mentioned in the section above, the output of the thresholding operation results in an image with two separate regions, one of them corresponding to the likely presence of the object and the other representing the background. In order to extract more information from these regions, it is necessary to carry out further processing of the image. Operations like edge detection or boundary finding are common to many image

processing algorithms.

3.1.2 Boundary Detection

Boundary finding and edge detection are theoretically two different problems; however, in this thesis, they are often used interchangeably. It is important to understand the meaning of an an edge with respect to image processing. A simple definition of an edge is a sudden change in the brightness level (Horn 1986, page 161). However, ignoring the texture information associated with each image and focusing only on the gray level contents, then, an edge may occur in essentially two ways: (Dougherty and Giardina 1987, page 74).

1. A border between two different regions, where each region is a homogeneous one with respect to some homogeneity criterion, may be termed as an edge.
2. A thin, dark arc on a bright background or vice versa may be termed as an edge.

The objects used in this thesis produce an edge as described in 1. above as it marks the transition from two regions of distinct intensity levels. The uniformity of some kind between these two sections of the image can be effectively exploited using differential operators that look for a gradient in the intensity. There are a variety of commonly used differential operators. The performance of any edge detector depends on the noise in the image and the corresponding blurring it causes at the edges. Work done in the area indicates that there is a trade off between the sensitivity and the accuracy of details. In general, shorter edges require more gradient or intensity change than the longer ones for successful detection. The process of edge detection is complimentary to that of image segmentation as both of them result in regions in the images based on a certain criterion. A differential operator in image analysis has a direct analogy to that *difference operator* encountered in basic calculus.

One of the simplest types of gradient or differential operators are called the *partial differential operators.* They simply calculate the rate of change of gray level in the horizontal and the vertical direction (Dougherty and Giardina 1987, page 74). Eq. 3.1 & 3.2 give the formal mathematical definitions of these operators.

$$[DX(f)](i,j) = \begin{cases} f(i,j) - f(i-1,j), & \text{if } f(i,j), f(i-1,j) \exists \\ 0 & \text{otherwise} \end{cases} \tag{3.1}$$

$$[DY(f)](i,j) = \begin{cases} f(i,j) - f(i,j-1) & \text{if } f(i,j), f(i,j-1) \exists \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

These are analogous to the *partial derivatives* in calculus as they measure the digital rate of change of gray level in x as well as the y direction. In case of binary images, this operation results in zero and non-zero numbers throughout the images.

The main problem with the partial differential operator as described above is its dependence on the direction. For detection of curved edges of a disc or an ellipse, this can be a serious weakness. A more complicated kind of operator is then used for such applications. It is called as the digital gradient operator (Dougherty and Giardina 1987, page 80). Some of the other type of operators commonly used are Sobel (Dougherty and Giardina 1987, page 83) in which each of the two operators are treated as a gradient filters.

The thesis attempts to find the boundary of the object placed on a 16×16 array. The limited resolution of the array is responsible in restricting the performance of the edge detectors and as such there is little difference between their relative performances. A different type of operator is used in this thesis which is based on scanning operations. An image is scanned from each of the four directions, namely, top, bottom, left and right, and then resultant images in either cases are added together. The technique generate the boundary of the outermost points.

A visual description of the technique can be seen in Fig. 3.2

The top left image indicates the original image. The top center image indicates the image after scanning from the left hand side of the image. The first non-zero pixel encountered is taken to be the boundary point. Similar operation is performed from the remaining three directions, namely, right, top and bottom. The result of adding each of these boundaries can be seen in the rightmost bottom image.

A vast branch of image processing deals with images that have some known attributes. This prior information can be very useful in extracting information about the image. Todate, in most robotics workcells, the set of objects a robot has to manipulate, is known a priori. An efficient use of such information can lead to interesting extraction of features as explained below.

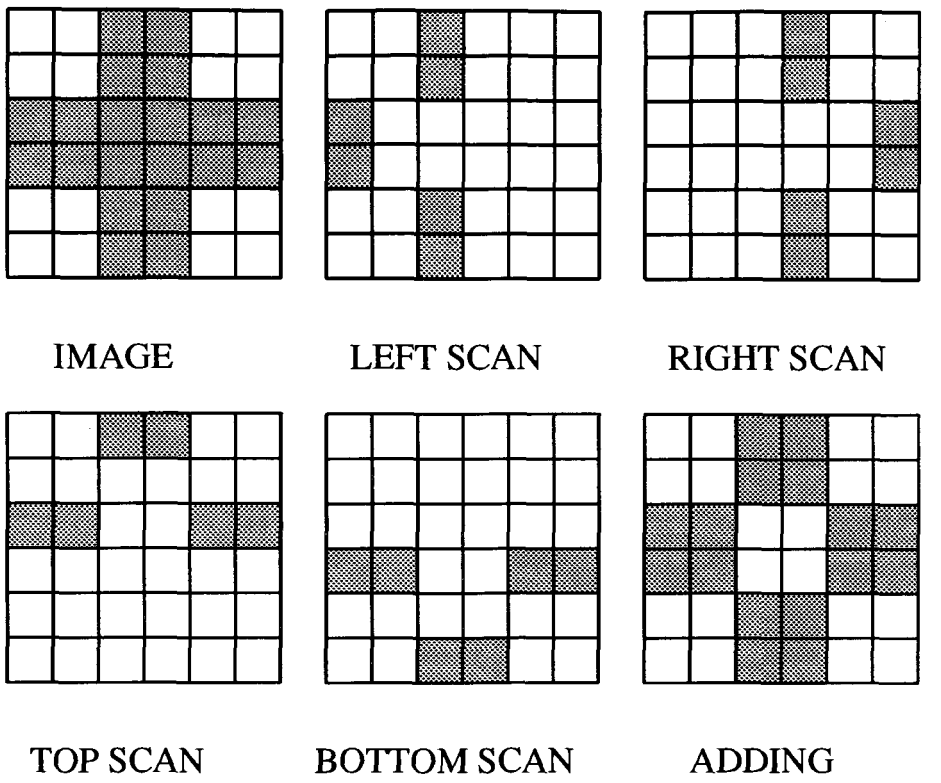


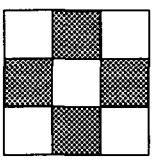
Figure 3.2: Description of the boundary finding algorithm

3.1.3 Mathematical Morphology

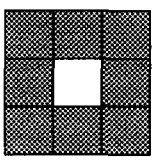
The term *morphology* in general scientific perspective means the study of form and structures and refers to analyzing images in terms of shapes and sizes. The process of shaping an image can be used to enhance its readability in terms of its features by mitigating the effects of noise in the original image. Morphological filtering presents a powerful set of operations that enables the user to incorporate *a priori* knowledge in picture processing. There are three fundamental morphological operations (Pratt 1991), namely, dilation, erosion and skeletonization: dilation expanding the image spatially, erosion causing a uniform shrinking and skeletonization resulting in stick figure representation. The following subsections present some of the relevant terms associated with morphological filtering that are used elsewhere in this thesis.

Connectivity

The morphological operations exploit the relationship between pixels that are deemed to be in the same class under some geometrical constraints. The geometrical constraints describe a way to relate or connect neighboring pixels. There are two basic classes of connectivity that are most commonly used: four connectivity and eight connectivity. Consider a simple figure as shown in the Fig. 3.3.



(a) 4 connectivity



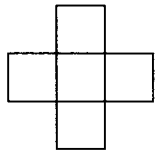
(b) 8 Connectivity

Figure 3.3: Connectivity Relationship

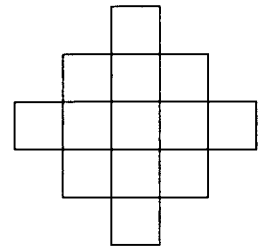
Fig. 3.3(a) indicates that the center pixel is said to be four connected if four neighbors in the North, South, West and East directions are all logical one along with the pixel at the center. As shown in Fig. 3.3(b); however; for eight connectivity, there are four more pixels that define the connectivity.

Structuring Element

The structuring element is a user defined structure that is used to shape an image as desired. A typical structuring element is an odd shaped, typically 3×3 mask, that is scanned over the image to dilate or erode it. The choice of an appropriate structuring element is an important design parameter as the overall outcome of the morphological processing is almost entirely dependent on it. As stated before, morphological operations allow the user to incorporate prior knowledge into processing techniques whereby develop an optimal processing strategy. In the context of this thesis, it is beneficial to exploit a priori knowledge about the set of objects being used for localization and identification purposes. The shape of the structuring element should thus be in close accordance to the expected image. Fig. 3.4(a) & (b) show some examples of structuring element that can be used to detect circular shapes. The evolution of these structuring elements will be explained in Chapter 4 where they are used.



(a) Structuring Element #1



(b) Structuring Element #2

Figure 3.4: Examples of typical structuring elements

Hit or Miss Transformations

The most basic of all morphological operations, erosions and dilations, can be performed by using simple hit or miss type of transformations. The images are considered to be binary images and therefore have values of "1" or "0" for any pixel. A simplistic example of such an operation can be described as follows. A structuring element of any shape is scanned over the entire image. The pixel under consideration always being the central pixel in the structuring element. The pixel in the output image corresponding to the central pixel is set to "1" or "0" depending on some relationship between the remaining pixels in the structuring element and the pixels underlying image. These operations can be mathematically expressed using what is called as Minkowski algebra (Dougherty and Giardina 1987). The operations of erosion and dilation are defined in the following (Jain 1989)

Consider a set X representing the object field and a structuring element B, both represented in their two-dimensional Euclidian space. If B_x represents the translated structuring element, with origin at x, then erosion of X by B is defined as the set of points x, such that, B_x lies entirely in the set X. Mathematically, the operation of erosion is represented by Eq. 3.3

$$X \ominus B \cong x : B_x \subset X \tag{3.3}$$

On the same lines, the operation of dilation can be defined as the set of points x such that B_x "hits" X. Mathematically, the operation is defined as shown in Eq. 3.4.

$$X \oplus B \cong x : B_x \cap X \neq \Phi \tag{3.4}$$

The operations of erosion and dilation can be applied in series to selectively remove random noise occurrence in the image and also shape the image. A simple example of noise removal can be seen in Fig. 3.5. The operation of erosion results in a much smaller image and removes random noise points; however, results in removal of some object points as well. The operation of erosion needs to be complemented using dilation. The complete operation is shown in Fig. 3.5.

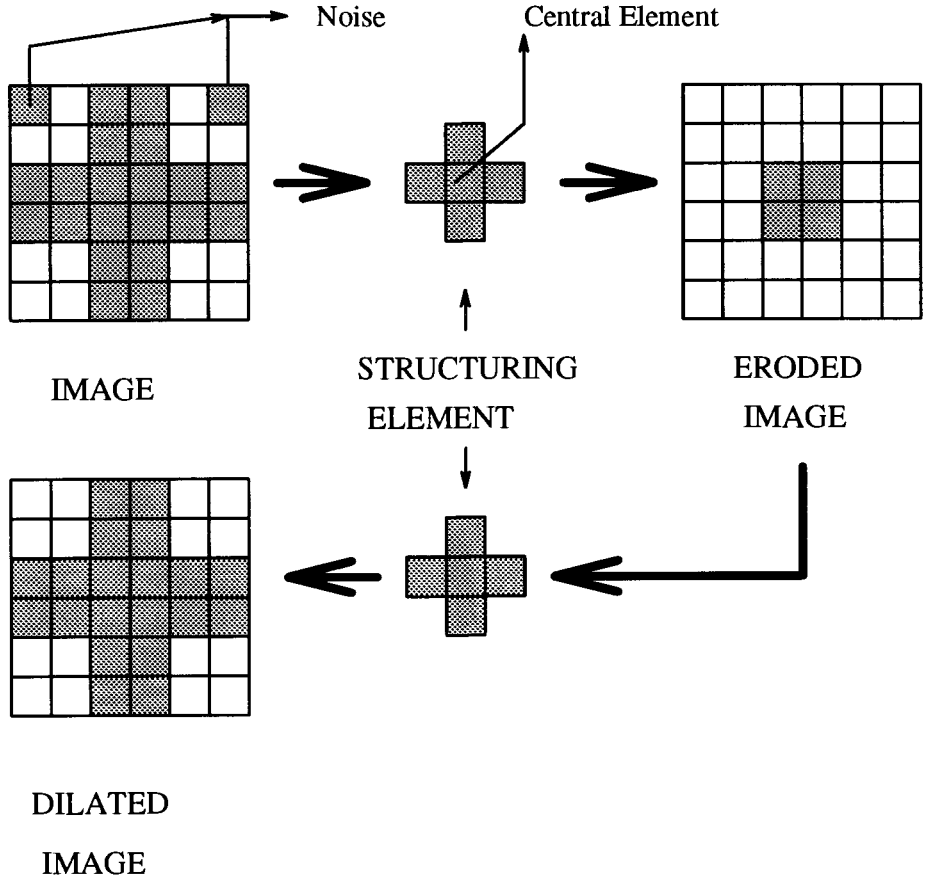


Figure 3.5: An example of noise removal using erosion and dilation

The operations of erosion and dilation can be performed iteratively depending on the application. However, depending on the cases, the process can saturate quickly and further iterations may not result in any change in the image.

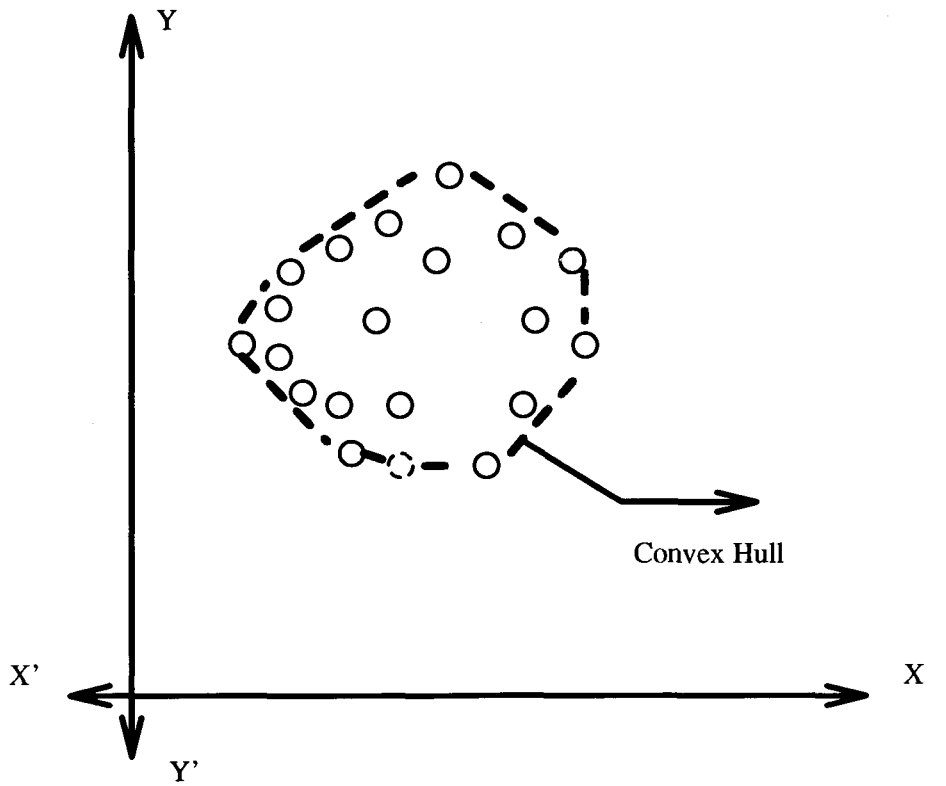


Figure 3.6: An example of convex hull

3.1.4 Convex Hull

The problem of finding a convex hull of a given set of points is a central one to many pattern recognition and image processing applications. It also finds vast applications in the field of computational geometry. By basic definition, a convex hull of a set of points in S is the smallest convex set containing those points. It can be easily visualized as follows. If there exists an elastic rubber band that is stretched to infinity and then released around any set S, then the shape the rubber band assumes after its release from infinity, determines its convex hull. An example of convex hull for a set of points can be seen in Fig. 3.6

Definitions of a few of the preliminaries associated convex hull can be found in appendix A.

A general convex hull algorithm used should compute the convex hull for a set of points in N dimensions. The set of points used in thesis are all 2-dimensional and as such the convex hull is relatively easy to compute. The problem of finding the convex

hull can be described in two parts as follows:

1. Convex Hull
   Given a set of N points in a set a set S, find the convex hull that is a complete description of the boundary associated with the convex hull.
2. Given a set of N points a set S , find the vertices of the convex hull.

Since the points described in this thesis are only 2-dimensional, the vertices of the convex hull gives a good indication of the boundary points. There is an abundance of literature describing a variety of algorithms that can be used to compute the convex hull given a set of data (Preparata and Shamos 1985). The main emphasis is on the effective use of the memory and the speed. A few of the important preliminaries need to be considered before looking into one of the typical algorithms used in practice.

Preliminary 1

A point p of a convex set is an *extreme* point if in two points $a, b \in S$ are such that p lies on the open line segment $\overline{ab}$.

Preliminary 2

A point fails to be an extreme point of a convex set S only if it lies inside a triangle whose vertices are in the convex set but is not itself one of the vertex.

Preliminary 3

A ray emanating from an interior of a bounded convex figure intersects the boundary in exactly one point.

Preliminary 4

Consecutive vertices of a convex hull occur in some ordered faction about some fixed point internal to the set.

A typical algorithm consists of the following steps:

1. Find the extreme points for a set.
2. Find any internal point.

3. Sort the extreme points by polar angle about the internal point found in 2.

The choice of the internal point can be trivial if we consider that the centroid of the set of points is internal to the system. However, the choice of the internal point varies from system to system. According to Graham's scan method (Preparata and Shamos 1985, page 100), the internal point is found by using any three non-collinear points. The speed of the algorithm is quite dependent on how fast one can sort the points about the internal point. There are a number of different algorithms used for this purpose. Some of the algorithms used for finding convex hull for any dimension use a technique often referred to as the QUICKHULL technique. The algorithm used in this thesis belongs to one such class of quickhull techniques and uses what is called as the beneath-beyond algorithm. The details of this algorithm can be found in Appendix C. The computational complexity of these algorithms is beyond the scope of this thesis and hence is omitted completely.

The set of vertices of the convex hull give an important indication of what points might lie on the convex surface that the object represents. An analysis of these vertices in terms of the extracting features can be performed using a standard data analysis technique. The data in which case being the coordinate values of the vertices of the convex hull.

3.1.5 Least Squares Fit

The technique of least squares fit is very common in data analysis and finds widespread applications for such purposes as curve fitting, fitting an equation to a given set of data points and so on. One of the widespread use of least squares fitting an be found in statistics, where prediction or an estimation from a set of random variables is a common practice. The prediction involves relating the past sequence of "happenings" to the future based on some criterion (Gersho and Gray 1992, page 84). A definition of the least squares operation can be found in different literature in different contexts. For example, in case of stochastic processes, let $X = (X_1, \ldots, X_N)^t$ be a vector of observed values and $Y = (Y_1, \ldots, Y_K)^t$ be the expected outcome vector. The prediction vector $\hat{Y} = \hat{Y}(X)$ should be chosen to be the "best" in some sense. Least squares estimate provides the most common measure of performance for such a predictor in terms of the mean squared error.

$$\epsilon^2(\hat{Y}) = E(\| Y - \hat{Y} \|^2) \equiv [(Y - \hat{Y})^t \cdot (Y - \hat{Y})] \tag{3.5}$$

$$\epsilon^2(\hat{Y}) = \sum_{i=1}^{K} E[(Y_i - \hat{Y}_i)^2] \tag{3.6}$$

Equations 3.5 & 3.6 represent the basic operations performed in least squares data analysis. The actual use of the technique is explained in a Chapter 4 where the technique will be used to compute the center and the radius values of circular objects.

3.1.6 Low Pass Filtering

Low pass filtering is a useful technique for noise suppression since many types of noises can be modeled as wide band processes. There is an abundance of literature available on the design and analysis of 1-dimensional filters in both the analog and the digital domain. The selection of the filter parameters depends on a variety of requirements including the sharpness of the response, allowable amount of ripples in the pass band as well as the stop band and gain etc. The straightforward extension of the 1-dimensional filter to 2-dimensions is possible in case of images.

A brief operational detail of the low pass filter is presented below. The operation of a simple 2-dimensional low pass filter can be explained with the concept of a window. Suppose, we create a 3×3 window with the center pixel being the pixel under consideration. Then, low pass filtering operation essentially computes the average of all the pixels in the window and replaces the value of the center pixel with the average. The effect is an obvious smoothing of the image. The intensity levels become more uniformly distributed throughout the image. This operation also helps in removal of random high frequency noise in the image. The disadvantage is the loss of details in the image. The details, being high frequency in nature, get lost in the process and hence features such as the edges and boundaries get smoothed out and lose their sharpness. Besides, a further disadvantage is the increase in the size of the actual image as compared to the original image. The performance of the low pass filter can be changed by modifying the size of the window and attaching different weights to the pixels summed. Mathematically, the operation can be defined as follows:

$$lpf[i][j] = W_{(i,j)} * image[i][j] \tag{3.7}$$

where $W_{(i,j)}$ represents the window over which the filter is defined and $*$ represents convolution.

3.2 Summary

The above section presented the reader with the background information on the image processing techniques used in the thesis. Use of mathematical morphology is new in the field of tactile sensing and hence a detailed description of the technology is presented. The next chapter describes how each of these techniques can be applied in our case. The results at every stage is also presented.

Chapter 4

Heuristic Processing

In the previous chapter, the reader was presented with background information on various image processing techniques employed in this thesis. Techniques like thresholding, morphological filtering, edge detection, low pass filtering, convex hull and least squares fit were discussed. In the following sections, we describe how each of this techniques can be applied to extract information from the array. It is important to note that the elements of the array provide 3-dimensional information, namely, the pressure and the x & the y position of the pressure acting on the array. The pressure is recorded in terms of voltages. For a visual display, the voltages are converted into the corresponding intensity values.

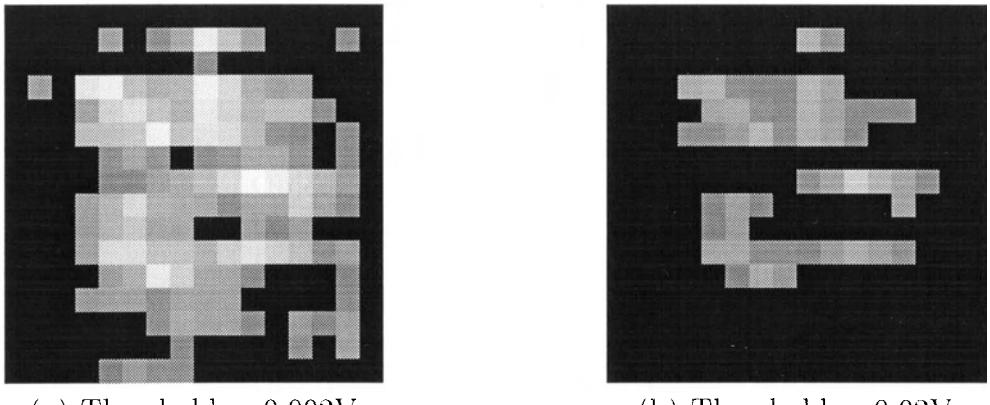
The following sections present the details of the processing methods employed to interpret information from a set of objects with varying size, weight, area and shape. The chapter starts by describing the technique used to determine the threshold value that is used for processing described in this chapter. Then, some of the results with localization of circular objects are presented. An extension to rectangular objects is described in the next section. The chapter then presents the development of an object identification system based on the array. Determination of the weight and area for an object is described in this section.

4.1 Determining the Threshold

Thresholding the input image forms the first block for many processing algorithms discussed in this thesis. The main reasons for employing thresholding as the starting

block can be discussed as follows. The data obtained from the hardware circuitry is stored in the digital form using a 12 bit analog to digital converter that is capable of storing values from -5V to $+5$V. The expected variation in the output values in our case is only from 0 to $+5$V which means that the useful resolution is reduced to 11 bits. This still means that the A/D converter is capable of recording values as low as $\frac{5}{2^{11}}V$. Because the A/D card is capable of recording such low voltage values, even a small residual voltage at an element gets registered as some positive voltage inspite of being very close to zero. Under these circumstances, it becomes important to mark some logical boundary between the points that do contribute some useful pressure information from those that do not. Thresholding operation achieves exactly the same purpose. By appropriately defining the characteristic function of the threshold operation, the boundary between the two values can be aptly defined.

The choice of the threshold value depends on a number of parameters and some of the methods of determining an appropriate threshold level were discussed in Chapter 3. It was also mentioned that the method to choose the most suited threshold is quite often application specific, which happens to be the case in this thesis. The thesis tries to interpret the pressure information obtained from the footprints of various objects placed on the FSR array. The choice of the threshold thus becomes quite important as it directly affects the points that get registered as a part of the footprint. Fig. 4.1 shows the footprint of a disc (weight=1860gm, radius=6.9cm) with two different threshold values.



(a) Threshold = 0.002V (b) Threshold = 0.02V

Figure 4.1: Footprints of the disc (weight=1860gm, radius=6.9cm) for two different threshold

As seen from Fig. 4.1(a) & 4.1(b), a very low value of the threshold (corresponding to 0.002V) results in a footprint that is very big whereas the higher value of the threshold (corresponding to 0.02V) results in a smaller footprint that has a lot of void points in places where the object actually makes a contact with the array. Another important feature to note here is that the void points are present in the footprint even for a low threshold value (corresponding to 0.002V) which is in accordance with our hypothesis, presented in chapter 2, of uneven contact surface and presence of bumps on the FSR array surface. From Fig. 4.1, it can be safely concluded that the threshold value directly affects the registered footprint.

The footprint of an object depends on the area and the weight of the object. In other words, the determination of the threshold should depend on the weight and the area of the object and should vary in some predefined manner for the variations in either or both of them. The argument presents a concept of employing a threshold that varies with the recorded data. The technique is often referred to as the *variable thresholding technique*. The variable threshold technique provides a method to incorporate the dependencies which vary the object footprint. The variable threshold technique obviously results in a sophisticated system which is likely to give better results than a simple universal threshold scheme in extreme cases. On the other hand, a universal threshold for all the objects is quite simple to implement and use and furthermore, results in considerably less computational overheads and burden on the system. Also, features such as the weight and the area need not be known a prior to determining the threshold value. To decide between the variable thresholding technique and a simple thresholding scheme, an experiment, described below was performed.

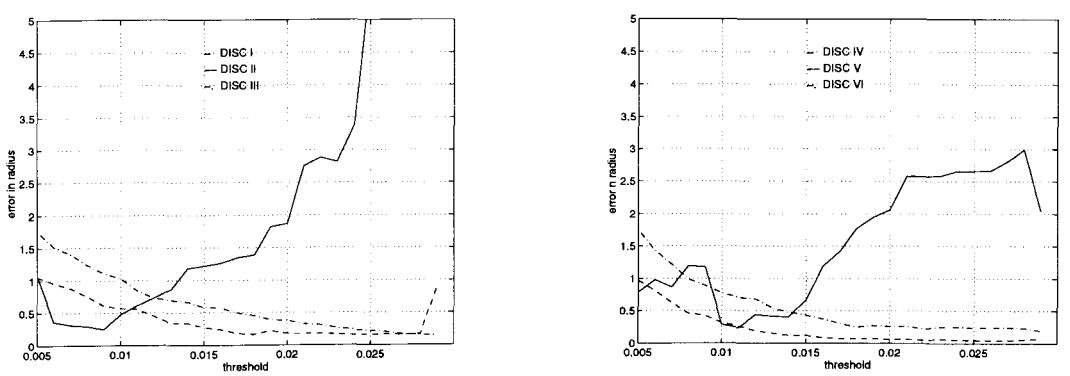
Table 4.1: Test objects used to determine threshold

Object	Weight(gm)	Radius(cm)
Disc I	1860	6.9
Disc II	2950	6.9
Disc III	3860	6.9
Disc IV	675	4.3
Disc V	1955	4.3
Disc VI	2675	4.3

The experiment was performed on a set of 20 readings taken for each object mentioned in Table 4.1 (We will prove later that the value of N=20 is a sufficient number

of readings). The radius for each measurement was computed using the technique described in Section 4.2 for a set of different threshold values. The threshold values were made to vary form a low to a high value. An error was calculated between the true value as given by the table and that found by the algorithm for every value of the threshold. These errors were plotted against the threshold values to indicate the "best" threshold or the "range of best" threshold values most suited for the objects described in Table 4.1. At this stage, it is important to note that the test objects described in the table are carefully chosen to provide the variations in area as well as the weight. To be more specific, the sets of Disc I,II & III and Disc IV, V & VI have the same radius (hence, the same area) but increasing weights respectively. This enables a comparison between objects of different weights but the same area and for different areas. The plot of the error versus the threshold values will indicate if the variable threshold is really a necessity.

In keeping with the above discussion, it is a better idea to first plot the results for Disc I, II & III and Disc IV,V & VI on separate graphs and then plot a combined set. Fig. 4.2 shows the graphs indicating the effects of varying the threshold values on the set of objects. The higher and the lower limits on the threshold values are heuristically chosen, after some experimentation, to be equal to 0.005V and 0.03V respectively.



(a) Set 1 radius = 6.9cm (b) Set 2 radius = 4.3cm

Figure 4.2: Effect of varying the threshold values on different objects

Fig. 4.2(a) & (b) indicate that, for the same area, heavier objects result in lower errors for higher threshold values. Another point to note is that the magnitude of the error for heavier objects is lower compared to the lighter objects at all threshold

values greater than 0.015V. In fact, for heavier objects, the magnitude of errors is less than 1cm for most part of the threshold variations. It is important to remember that the size of each pixel is effectively 1.3cm × 1.3cm and that the error of less than 1cm represents a reasonable accuracy of within one pixel. Finally, in either case, the lighter objects show larger errors for threshold values in excess of 0.015V. A combined plot of the errors as shown in Fig. 4.3, is now plotted to compare the errors for both the sets or to compare the result of varying areas for varying weights.

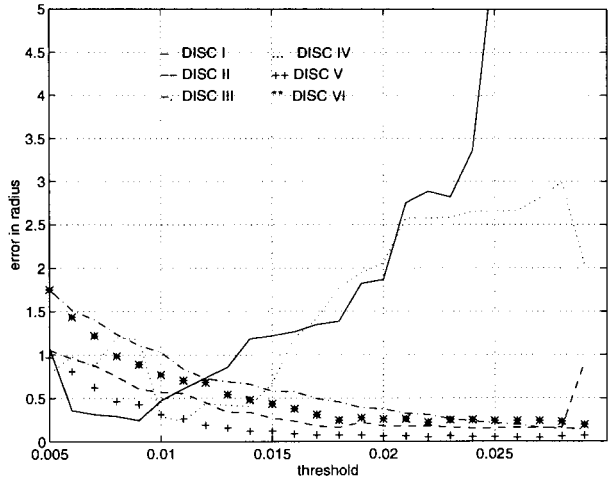


Figure 4.3: Effect of varying the threshold values on the error in radius measurement

The figure shows that the errors are found to be the lowest over a range of 0.01V to 0.015V for all the objects. The maximum average error over this range of threshold values is found to be less than 1cm. For all practical purposes, any threshold value between 0.01 to 0.015V performs equally well. We have chosen a value of 0.01V, the lower of the two values, in order to minimize the information lost for the lighter objects. The graph also conclusively proves that there is no compelling need for a variable threshold value as a universal threshold value of 0.01V seems to be a reasonable choice for the set of objects. The value of 0.01V is used as the threshold value from now on in this chapter unless specifically mentioned.

At this stage, it still remains to prove that the choice of 20 readings for each object is reasonable. An experiment described below, was performed to arrive at the value of 20.

A graph of variance in the error reading for different values of N was plotted and N

corresponding to the "flat" region of the curve was chosen as the minimum number of readings. In other words, the value of N was chosen in such a way that the variance in the error is not significantly affected by reasonable increases in N. Ideally, the variance value $\rightarrow \Delta$ as N $\rightarrow \infty$.

The graph corresponding to the threshold value of 0.01V different values of N is as shown in Fig. 4.4.

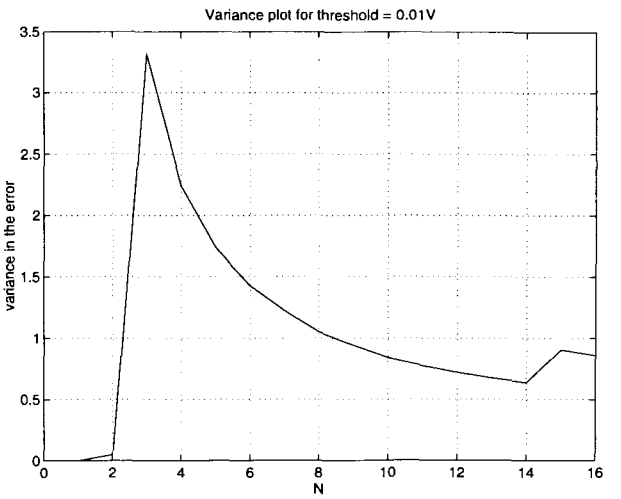


Figure 4.4: Varying N (total number of readings) for a threshold of 0.01V

It can be seen that the value of variance is quite low (approximately 0.5) after N of 14. For all practical purposes, it can be safely said that the value of 20 is a safe upper limit on the total number of readings that one should take.

We finally present one plot, shown in Fig. 4.5, that indicates the combined effects of variances of the errors, N and the threshold values. This plot shows variances over the entire range of threshold values for all values of N. The plot further strengthen the argument that the value of N=20 is indeed the sufficient.

The above discussion showed that a universal threshold performs satisfactorily in our system and the value of 0.01V as the threshold seems to be reasonable choice for thresholding.

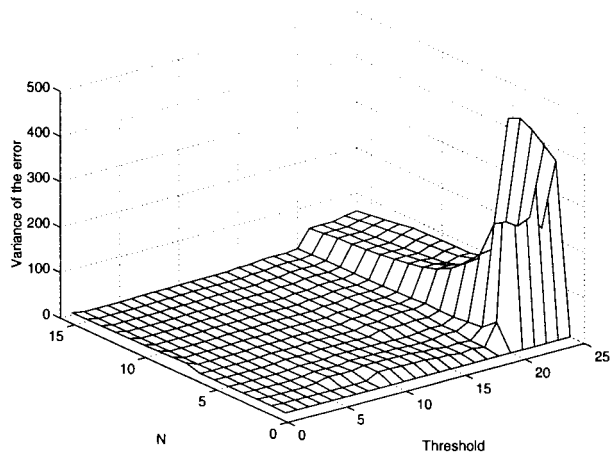


Figure 4.5: Combined plot for number of readings, variance of the error and the threshold

4.2 Localization Of Circular Objects

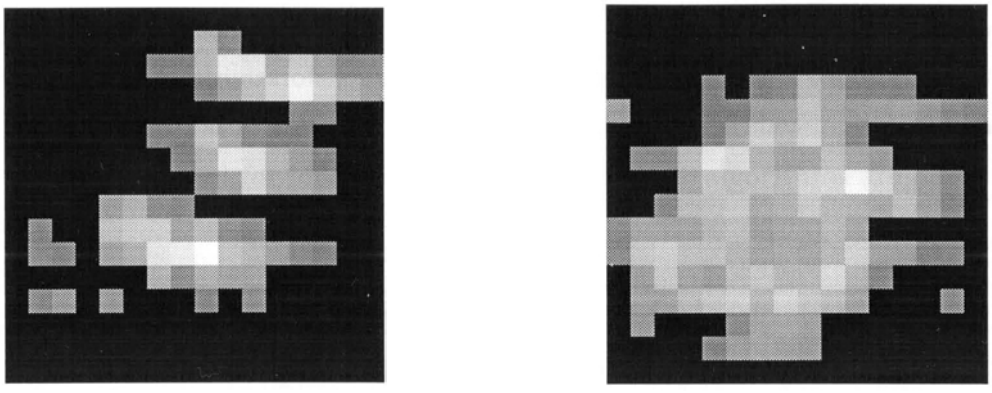
The thresholding operation described in the previous section, by itself, does not provide much information about the object in terms of its weight, area or its location on the array. It simply indicates the points that are deemed to be within the footprint of the object. In order to extract relevant details like the position, weight and the area of the objects, some further processing is necessary. In following subsections, we describe a method employed to localize circular objects by finding the centroids and the radii of the objects.

Why Circular objects?

Before proceeding onto describing the method to localize circular objects, it is necessary to discuss the reasons for restricting the shape of the footprints to be circular.

At this stage, it is important to remember that each element of the array is quite big in dimension (of the order of $1\text{cm} \times 1\text{cm}$). Because of such a large size, an absence of one element from the object footprint is likely to affect the readability of the image significantly. The problem of size of the element coupled with the unevenness of the contact can cause quite a lot of ambiguity in the shape of the recorded footprint. To emphasize this point, consider two images: a disc and a rectangle, as shown in Fig. 4.6.

As seen from Fig. 4.6, it is almost impossible to distinguish between a circular and



(a) Circular Object (b) Rectangular Object

Figure 4.6: Typical images for a circle and a rectangle

a rectangular footprint just by inspecting the recorded footprints. Hence, as an initial study, it was decided to choose geometrically the simplest class of the objects, namely, the circular class. The circular objects by definition do not have any orientation and hence the problem of locating an object belonging to circular class reduces to simply finding the centroid and the radius of the object.

Such a class of objects seems quite restrictive at a first thought. However, it is important to note that the case is considered as an initial study, and that in a latter section, the localization of rectangular objects is described. Further, in support of such a restricted class, it can be pointed out that the object classification and identification systems developed todate, all work on a restricted class of objects and some attributes of the objects are known a priori to the detection algorithm. Hence, the choice of circular objects as an initial class of objects does seem reasonable. The section that describes the object identification technique suggests another way of classifying the objects, namely, based on their weights and the area.

In the following subsection, we describe the block diagram of the localization algorithm. The subsequent subsection, describe each block of the algorithm in detail.

4.2.1 The First Localization Algorithm

The technique employed for the localization is block based with each block performing a specific task. The output of every block is a processed 2-dimensional image. A typical image is shown at the end of each processing step. The algorithm evolves in

three stages with each stage trying to build upon the results obtained from the last stage with some modifications. A complete sequence of images for each of the test object described in Table 4.2 is shown at the end of the third development stage. The set of test objects used for localization in all of the three algorithms are described in Table 4.2.

Table 4.2: Circular test objects used for localization

Object	Weight(gm)	Radius(cm)
DISC I	1860	6.8
DISC II	6360	5.1
DISC III	6810	7.7

The block diagram of the first stage of development can be seen in Fig. 4.7.

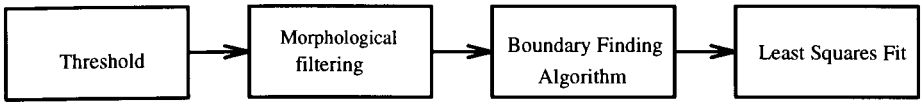


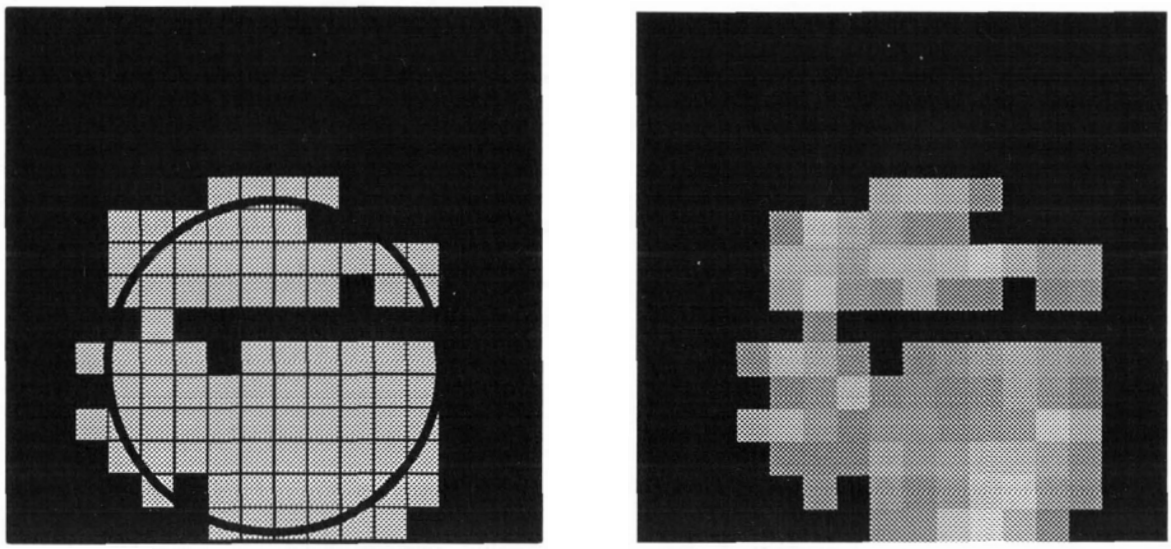
Figure 4.7: Block diagram of first attempt to localize circular objects

Thresholding

The first processing block is thresholding the image. As discussed in Section 4.1, a threshold value of 0.01 A typical image of DISC I (as described in Table 4.2) after thresholding is as shown in the Fig. 4.8. The circle on top of the figure indicates the exact position of the object on the array. This helps to visually locate the object against the obtained footprint.

4.2.2 Morphological Filtering I

The second block in Fig. 4.7 consists of filtering the data to remove noise points and to improve the readability of the image in terms of its shape. As mentioned in the previous chapter, there are three main steps involved in morphological filtering: the choice of the structuring element, eroding the image and dilating the image. The following subsections describe actual implementation of each of these techniques.



(a) (b)

Figure 4.8: (a)Actual position of the object (b)Image after thresholding

Choice of the Structuring Element

In the previous chapter, it was mentioned that the choice of the structuring element provides a powerful way to incorporate prior knowledge into image processing. In our case, the shape of the object footprint is known to be circular. Hence, the shape of the structuring element should be as close to a circle as possible. As a relevant note, it is important to remember the size of the original image, which is 16×16. The small size of the image quite drastically restricts the choice of the structuring element as explained below. Consider two elements as shown in Fig. 4.9

(a) Structuring Element #1 (b) Structuring Element #2

Figure 4.9: Structuring elements

It is clear that the second element is more circular than the first element, however, it is a lot bigger in size than the first structuring element. In some cases, an attempt to

erode the original image (size 16×16) using the element shown in Fig. 4.9(b) results in a blank picture as every element of the original image gets eroded after failing to meet the criterion. Dilation or any further processing is thus rendered useless. The size of the original image is thus quite critical in restricting the choice of the structuring element and, for the moment, we are limited to using the smallest possible "circular looking" element as shown in Fig. 4.9(a).

Erosion

Erosion selectively removes the noise from the image by searching for an exact match between the structuring element and the underlying image. A result of applying erosion to the thresholded image shown in Fig. 4.8 is shown in Fig. 4.10.

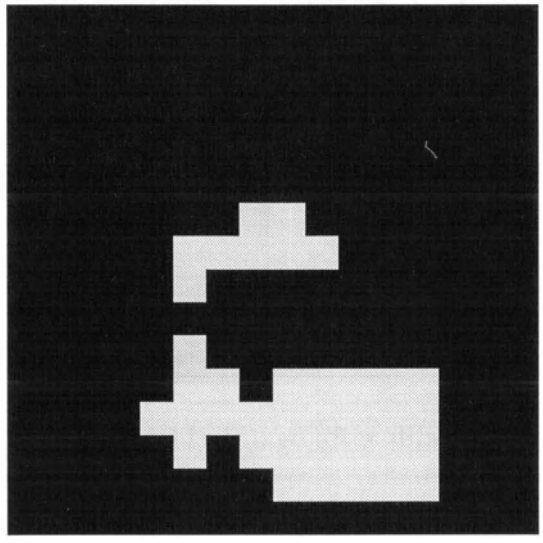


Figure 4.10: Image after eroding

In the previous chapter, it was said that erosion results in an image that is quite small compared with the original image which can be readily verified by glancing at the two images presented in Fig. 4.8 & Fig. 4.10. It should also be noted that an eroded image by itself conveys little or no information. In fact, it results in loss of information when compared with the original image. The real improvement can be seen only after dilating the image using the same structuring element.

Dilation

The process of dilation is logically opposite to that of the erosion and results in an image that is quite bigger in size than the eroded image. The main purpose of morphological filtering is to clean the image of random noise pixels and reshape the resultant image to the desired shape. The "cleanliness" of the dilated image directly depends on that of the eroded image, since the process of dilation follows erosion and uses eroded image as its input. Fig. 4.11 indicates the result of applying dilation to the image as shown in Fig. 4.10.

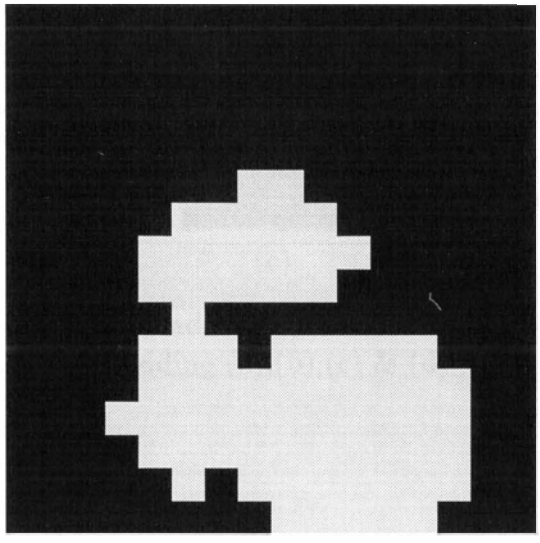


Figure 4.11: Image after dilating

It is interesting to note that the dilated image looks more circular than the original image and that some of the "voids" seen in Fig. 4.8 have been filled or reshaped. The "clean" image is thus ready for further processing.

4.2.3 Boundary Finding

A typical boundary finding algorithm finds the points that are deemed to be on the boundary of the image. In the algorithm used in this thesis, the points that belong to the extremities of the image are assumed to be on the boundary of the image. The size of the image is the chief restricting factor and as such the performance of most boundary finding algorithms would be the same. As explained in Chapter 3, the algorithm works in four main steps. The image is scanned from left, right,top

and bottom and the resultant four images are added together to generate a complete boundary description. In the left and the right scan, the image is scanned row-wise for every column. The rest of the row is completely ignored upon encountering the first non-zero element in any row. In the top and the bottom scan, the image is scanned column-wise for every row. The rest of the column is left untouched once a non-zero pixel is encountered in any column. The final image is formed by adding the result of each scanning.

Fig. 4.12(a),(b),(c),(d),&(e) show the result of scanning from left, right, top, bottom and finally adding the images.

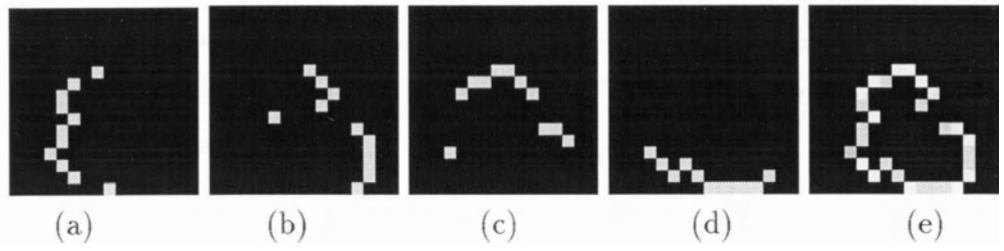


Figure 4.12: Result of boundary finding algorithm after (a)Left scan (b)Right scan (c)Top scan (d)Bottom scan (e)adding (a),(b),(c) & (d)

4.2.4 Least Squares Fit

The least squares fit algorithm uses a set of input point and fits the "best" circle that passes through these points. The term "best" is in the least squares sense. The input to the algorithms is the Cartesian co-ordinates of the points obtained after finding the boundary. The output of the algorithm describes the centroids and the radius of the object. The centroid position is mentioned relative to the top left hand corner of the array.

If N represents a set of boundary points represented by their Cartesian co-ordinates, (X_i, Y_i), then the distances of points (X_i, Y_i), for $i = 1, \ldots, N$, from a circle with radius R_b and center (X_b, Y_b) is given by

$$d_i^2 = (X_i - X_b)^2 + (Y_i - Y_b)^2 \tag{4.1}$$

and

$$d^2 = \sum_{i=1}^{N} d_i^2 \tag{4.2}$$

The error between the radius R_b and the distance shown in Eq. 4.1 can be written as

$$e = \sum_{i=1}^{N} (R_b - d_i)^2 \tag{4.3}$$

Eq. 4.3 is minimized iteratively to obtain the best estimate of the radius and the center points. The complete derivation of the least squares algorithm used to fit a circle to the set of points is described in appendix B.

The result of the algorithm is tested on a number of input data points and tabulated. Table 4.3 describes the results for the errors in the center and the radius measurements. The set also shows the maximum and the minimum errors obtained on a set of input data.

Table 4.3: Results for structuring element #1

	Xcenter(cm)		Ycenter(cm)		Radius(cm)	
	Max	Avg	Max	Avg	Max	Avg
Disc	0.4	0.3	2.0	1.3	2.2	1.2
Cyl I	1.1	0.7	0.6	0.4	2.3	2.0
Cyl II	1.5	0.6	1.5	0.9	2.6	2.0
Max=Maximum Error; Avg=Average Error						

The results are "reasonable", however, there is scope for further improvement especially in the measurement of radii for different readings. As mentioned earlier, the choice of the structuring element was quite restricted due to the size of the image. The second stage of development aims to get around this difficulty.

4.2.5 The Second Localization Algorithm

Improvements in the performances can be expected if a more circular structuring element is used; this is possible if the resolution of the image is increased. No improvements can be made on the actual set of data in terms the image size, since the the spatial resolution of the FSR array is fixed; however, the resolution of the array can be artificially increased by subdividing each element of the array into an $N \times N$ image. The process of subdivision is described in Fig. 4.13.

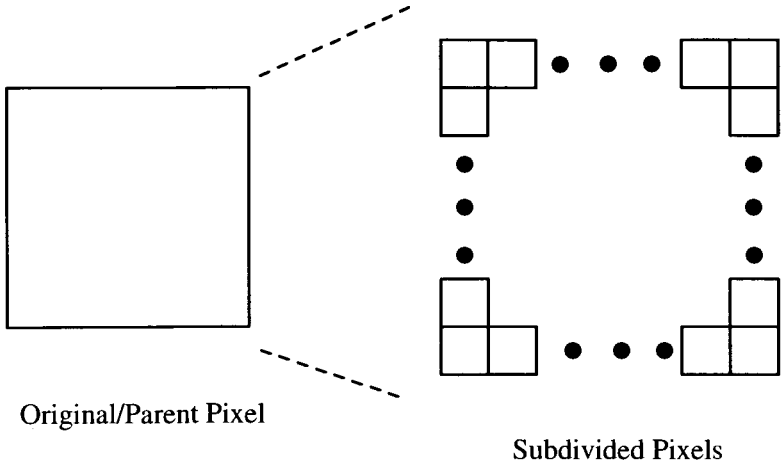


Figure 4.13: Concept of subpixel image processing

As shown in Fig. 4.13, each divided pixel inherits the the state of the parent element. In other words, the divided pixels are deemed to be 'ON' if the parent pixel is '1' and 'OFF' if the parent pixel is '0'. It is important to note that processing the divided image may result in part of the originally 'ON' element being absent from the footprint, or a part of the originally 'OFF' parent pixel to be 'ON'. This leads to extrapolating information about the shape of the object. However, an obvious limitation of the process is the fact that originally there is just one value for an entire parent element. In other words, after thresholding the image, the only information that we have about the parent element is whether its 'ON' or ''OFF'. Thus to begin with, the image size was only quadrupled in area by subdividing each parent pixel of the original image into a 2×2. The size of the resultant image thus increased to 32×32 from 16×16. The block diagram of the second process thus gets modified as shown in Fig. 4.14. The darker block represents the addition to the block diagram as seen in Fig. 4.7

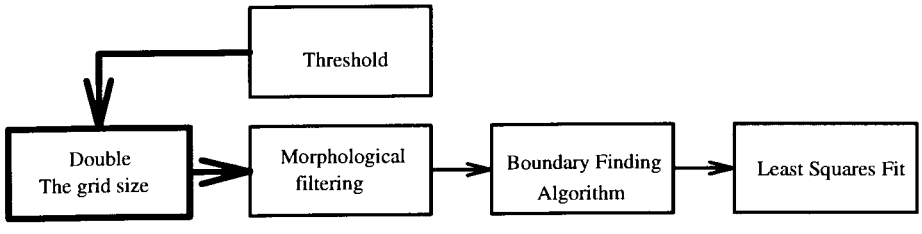


Figure 4.14: Block diagram of the second localization algorithm

The entire operation is repeated on the starting image shown in Fig. 4.8.

4.2.6 Morphological Filtering II

The structuring element as depicted in Fig. 4.9(b) is used for erosion and dilation purposes. The images after eroding the doublesized replica of image shown in Fig. 4.8 can be seen in Fig. 4.15.

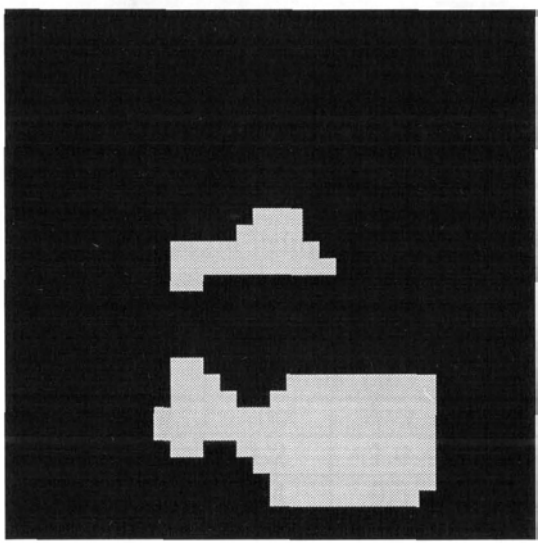


Figure 4.15: After erosion using structuring element #2

Comparing the Fig. 4.10 & 4.15, it can be seen that the second image has more points compared to the first one. The pronounced effects can be seen at the edges which appear to be more circular like than the first image. The image is then dilated using structuring element as shown in Fig. 4.9(b) and the result is shown in Fig. 4.16.

The figure definitely shows some improvement over that seen in Fig. 4.11. The edges seem more circular and more "zig-zaggy", as would be expected of a circular image. The boundary finding algorithm as described above is applied on the image shown in Fig. 4.16. The result can be seen in Fig. 4.17.

The least squares fit algorithm as described before is applied to the boundary points to find the centroid and the radii values. The result obtained using the above approach is indicated in Table 4.4

Comparing Tables 4.3 and 4.4, the results for the radius has improved somewhat, while as those for the centroids seem to be approximately the same. It can be seen

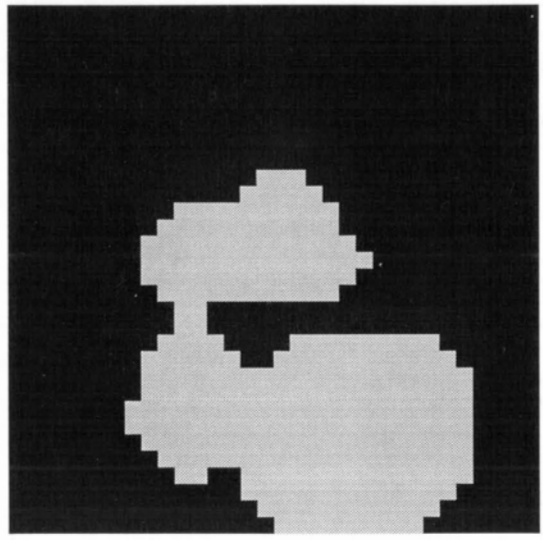


Figure 4.16: After dilation using structuring element #2

Table 4.4: Results for structuring element #2

	Xcenter(cm)		Ycenter(cm)		Radius(cm)	
	Max	Avg	Max	Avg	Max	Avg
Disc	0.8	0.5	1.2	1.0	1.3	0.8
Cyl I	0.5	0.4	0.4	0.3	2.0	1.6
Cyl II	2.0	0.8	1.4	0.7	2.2	1.7
Max=Maximum Error; Avg=Average Error						

that the improvement provided by the subdivision is far from dramatic. However, further improvements can be expected by generating a better boundary description for the "circular" objects. This is the third and the final development stage for the localization of circular objects.

4.2.7 The Third Localization Algorithm

As mentioned before, the presence of dirt, bumps or an uneven contact surface cause an improper registration of the object footprint. Typically, elements which actually belong to the footprint are found to be void. Under such a condition, the boundary generated by using the boundary finding algorithm often fails to be close to circular. Fig. 4.18(a) illustrates the object footprint and Fig. 4.18(b) illustrates the boundary found by the algorithm.

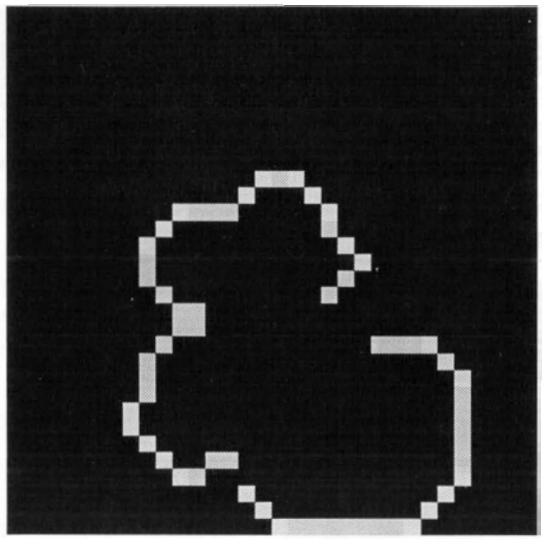


Figure 4.17: After finding the boundary

A least squares fit to such a boundary obviously generates bad results for the radius values. The occurrence of such a case can not be ignored as a stray bad case, since there is no control over the appearance of noise pixels in the original image. For a better approximation, it is thus necessary to generate a better set of boundary elements for the least squares algorithm. At this stage, the prior knowledge about the circular nature of the object footprint can be further utilized. An operator that selectively retains only those points that lie on a supposedly circular footprint can be applied to the boundary found using the above mentioned algorithm. A circle is a convex surface and hence finding the convex hull of the boundary typically results in a more circular boundary. The block diagram of the third and final stage for the localization is thus illustrated by Fig. 4.19. As before, the addition to Fig. 4.14 is shown by a thicker block.

4.2.8 Convex Hull

The actual convex hull algorithm used is due to Bradford Barber, Dobkin, and Huhdanpaa (1993). A brief description of this algorithm can be found in Appendix C. The result of applying the technique to the boundary as illustrated by Fig. 4.18(b) is shown in the Fig. 4.20.

The Fig. 4.20 indicates only the vertices of the convex hull. The actual convex

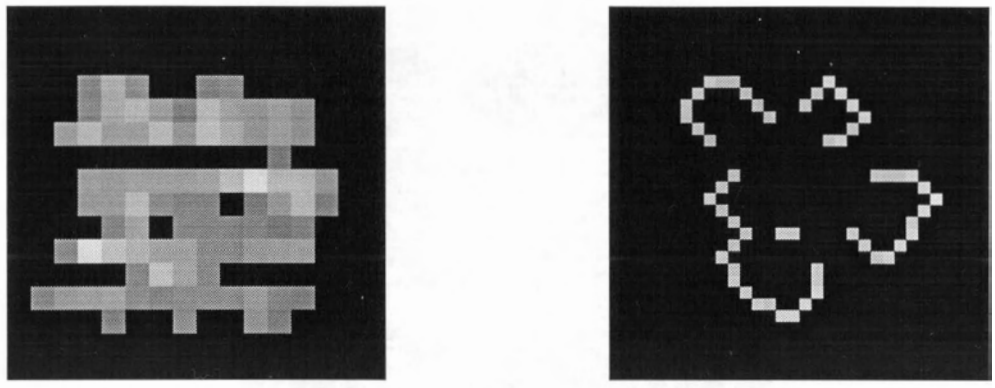


Figure 4.18: A typical failure case for boundary algorithm (a) A typical unprocessed image (b) Boundary of the image in (a)

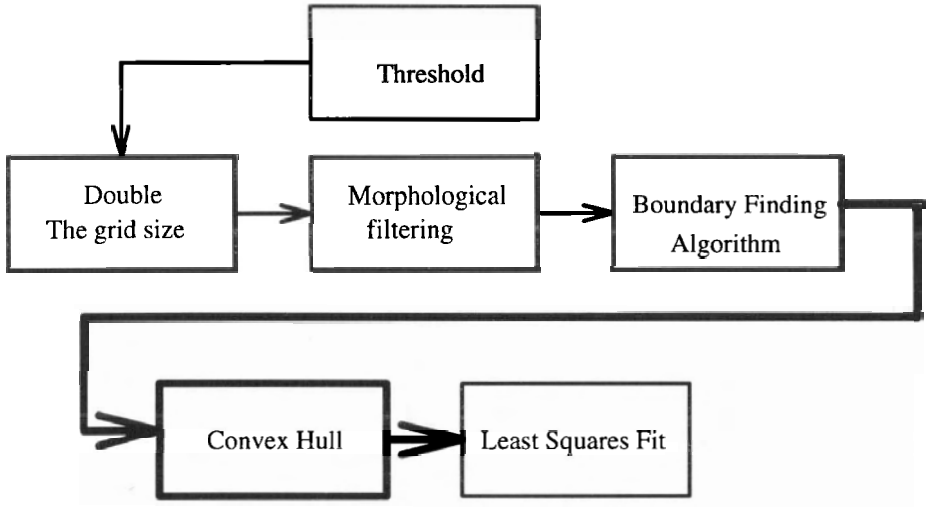


Figure 4.19: Block diagram of the final localization Algorithm

hull can be constructed by joining these vertices. It should be noted that the least squares fit algorithm uses only the vertices of the convex hull.

The results after using convex hull technique can be seen in Table. 4.5.

Comparing Tables. 4.3 to 4.5, it can be seen that there is a dramatic improvement in the quality of the radius. The centroid values are also somewhat improved. Importantly, the errors are now typically less than or around 1cm, which is less that the size of one element on the array. The Fig. 4.21 represents the sequence of images that are generated for each object in the Table 4.2 after each block of the block diagram shown in Fig. 4.19.

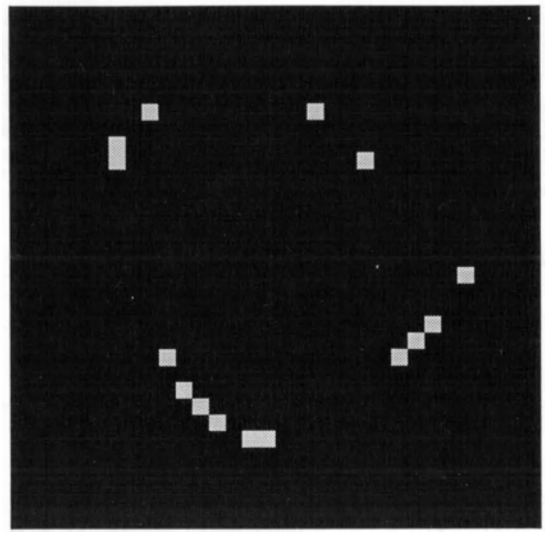


Figure 4.20: Convex hull of the bad boundary

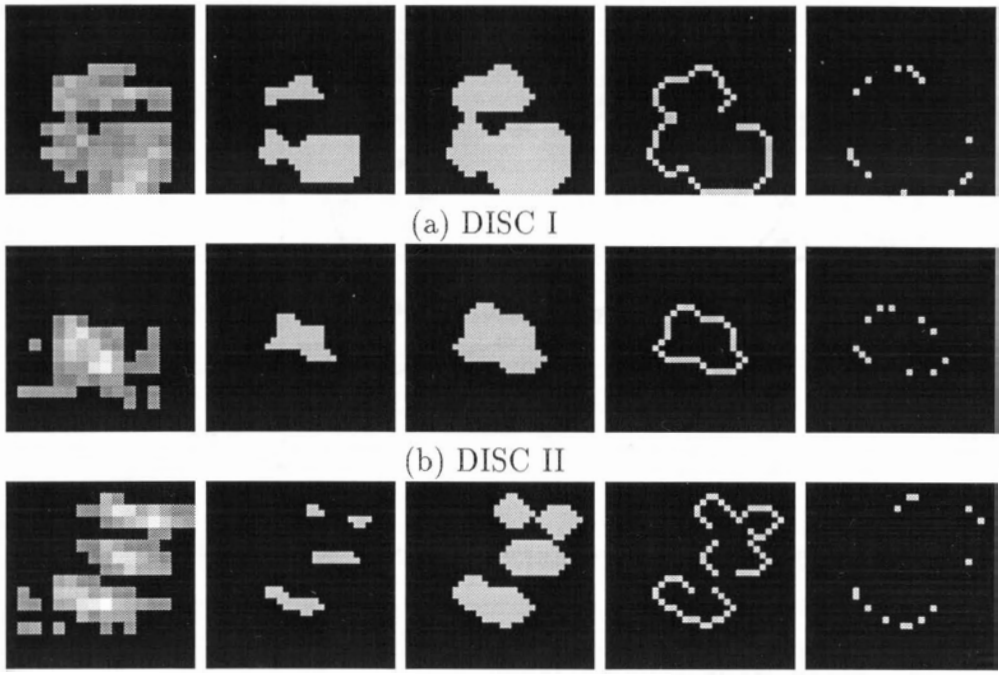
Table 4.5: Results after convex hull

	Xcenter(cm)		Ycenter(cm)		Radius(cm)	
	Max	Avg	Max	Avg	Max	Avg
Disc	0.9	0.6	1.0	0.6	0.4	0.3
Cyl I	0.7	0.4	0.6	0.5	1.3	1.0
Cyl II	0.7	0.5	0.6	0.4	0.7	0.4
Max=Maximum Error; Avg=Average Error						

In the next section, we give an extension of the above technique to localizing objects with rectangular footprints.

4.3 Localization of Rectangular Objects

The localization of rectangular objects poses the additional problems of determining the orientation of the object along with its length and the breadth. The problem is compounded by the fact that the starting image is sensitive to the random "noise" present due to bumps and uneven contact surfaces. A look at Fig. 4.6 shows that there is very little distinction in terms of the object footprints. In this section, we assume a prior knowledge that the object is a rectangle. The routines developed follow the same logical flow as that for the circular objects explained in the last section. The object attributes like the length and the breadth are found by fitting an ellipse to the



(a) DISC I

(b) DISC II

(c) DISC III

Figure 4.21: Sequence of images after each step in the algorithm. From left to right, original image, after erosion, after dilation, after boundary finding and after convex hull

image. In the next subsection, an argument for elliptic fit as opposed to a rectangular fit is presented. This is followed by the two step evolution of the algorithm to localize elliptic objects. The section concludes by presenting some motivation behind the object identification system.

4.3.1 The Elliptic Fit

In the last paragraph, it was mentioned that an ellipse is fit to the rectangular footprint. The reasons for this are as follows. An ellipse is a continuous geometrical structure as opposed to a rectangle, which has discontinuities at the vertices. A rectangle placed at the origin can be described by 4 lines of the form; $X = a; X = b; Y = c; Y = d$, where a, b, c, d are constants. It is a difficult structure to account for analytically. On the other hand, the equation of an ellipse centered at origin is of the form $\frac{X^2}{a^2} + \frac{Y^2}{b^2} = 1$, where $2a, 2b$ are the lengths of the major and the minor axes respectively. The equation reduces to that of a circle if $a = b$. The entire

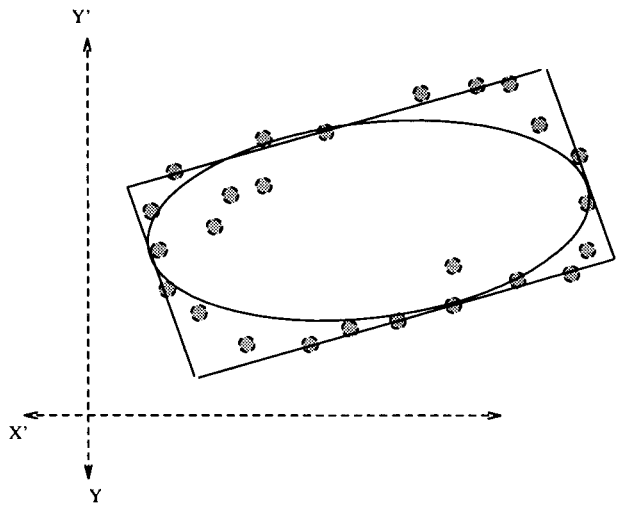


Figure 4.22: Fitting a rectangle and an ellipse

last section was devoted to localization of circular objects and hence fitting an ellipse is just an extension of the technique explained in Appendix B but with two variables, namely; the lengths of the major and the minor axes, instead of just the radius. It can be argued that there might be marginal improvement in extreme cases by fitting a rectangle instead of an ellipse; however, the complexity of the analysis is greatly increased and the consistency with the previous methods is not maintained. Fig. 4.22 illustrated a typical case where an ellipse and a rectangle are fitted to a set of points.

The figure indicates that there is little gain by replacing elliptic fit by a rectangular fit. In the next subsection, the first block diagram for localization of rectangular objects is described followed by in depth description of each of the blocks in the diagram.

4.3.2 The First Localization Algorithm for Rectangular Objects

Rectangular objects are more difficult to describe in terms of length and the breadth information because of the added feature of the associated orientation. As before, we start the section by describing the test objects used for the rectangular object localization. The algorithm was tested on two objects of different dimensions as described by Table 4.6. The objects were placed randomly on the array with the only precaution being that the entire object lies on the array and no part of it is outside the

array. The block diagram for the first approach of localization is depicted in Fig. 4.23

Table 4.6: Rectangular test objects used for localization

Object	Weight(gm)	Length(cm)	Breadth(cm)
RECTANGLE I	3000	16.0	7.0
RECTANGLE II	1500	14.0	3.0

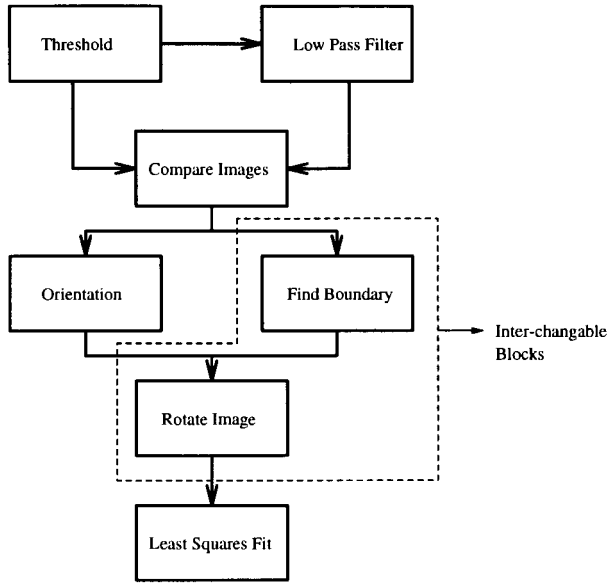


Figure 4.23: Block diagram for the first localization algorithm for rectangular Objects

In the following subsections, we present a detailed description of each block and its purpose in the block diagram.

Thresholding

Thresholding the image serves the same purpose as described before. It acts as a preliminary filter that simply describes the lowest permissible level that is non-zero. An image of RECTANGLE II described in Table 4.6 after thresholding can be seen in Fig. 4.24

In the next section, we digress a little bit from the methodology used for circular object localization.

As seen from Fig. 4.24, the image after thresholding still has some random noise points that do not belong to the original image footprint. The image also has an

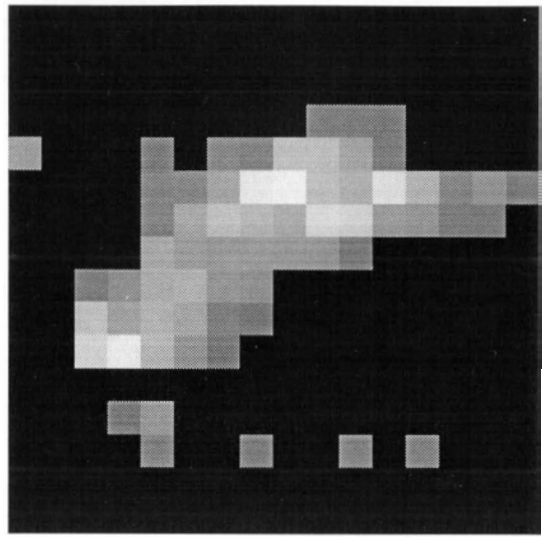


Figure 4.24: Image for RECTANGLE II after thresholded

orientation associated with it. A direct extension of the circular fit algorithm would lead to a different morphological operator for every orientation or at least a group of orientations. This is quite difficult due to the obvious limitation on the choice of the structuring element, imposed due to the total number of array elements. However, the number of morphological operators reduces to one if the image is rotated such that the major axis of the image coincides with the X-axis and the centroid of the image is at the origin. In other words, the shape of the image is predictable after rotation in the above manner and hence the design of the structuring element is the same for all the images. In fact, this forms the basis for the approach mentioned in Fig. 4.23. The image assumes a known shape once the mean has been removed and it has been rotated about its major axis. The first step in the processing approach thus becomes finding the orientation of the image.

4.3.3 Low Pass Filter (LPF)

In the previous paragraph, it was argued that the direct use of morphological filtering on rectangular footprint is quite difficult and for all practical purposes, close to impossible. Hence, some alternative techniques have to be implemented to eliminate the noise from the image. A simplest technique is to use low pass filtering on the image. As mentioned in the theory for low pass filtering in the last chapter, the main

advantage of low pass filtering is that it removes isolated noise points. It was also mentioned that the main drawback of this technique is a dilation of the image. The image shown in Fig. 4.24 after low pass filtering is illustrated in Fig. 4.25 low pass filtered.

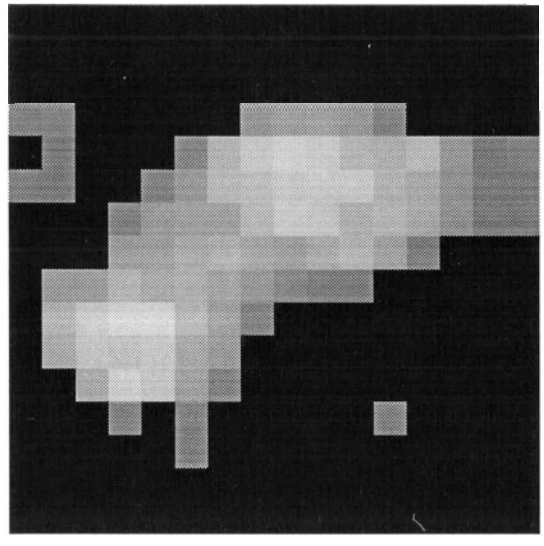


Figure 4.25: Image for RECTANGLE II after low pass filtering

Comparing the two images shown in Figs. 4.24 & 4.25, it is seen that the low pass filtered image is much bigger than the original image. Also, that a lot of isolated noise points have been eliminated.

Compare Image

Low pass filtered image is a smoother and a bigger version of the original image. However, a careful inspection of the images in Figs. 4.24 & 4.25, suggests that the dilation of the image affects the only the periphery of the image. Hence, to retain the size of the original image, a pixel by pixel comparison of the two images, namely, the thresholded and the low pass filtered, is performed. Only those pixels which are common to both the images are retained and all the other pixels are ignored which resembles "logical ANDing" operation. This results in retaining the original size of the thresholded image and at the same time, get rid of a lot of the noise points associated with the original image. The resultant image after comparing images in Figs. 4.24 & 4.25, is depicted in Fig. 4.26.

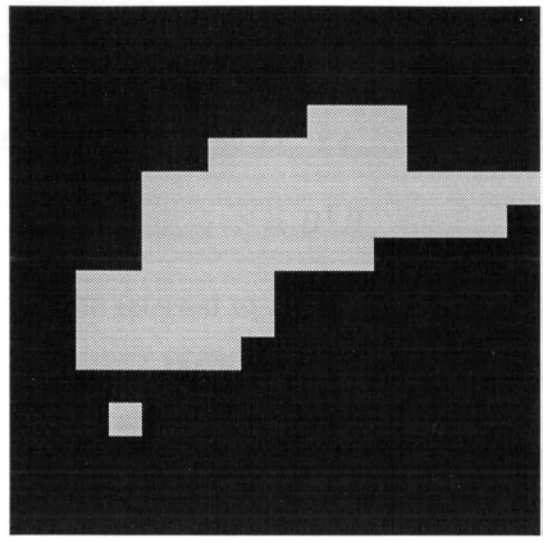


Figure 4.26: Image for RECTANGLE II after comparing the thresholded and the low pass filtered images

The image is not only free of some of the noise points but also has the same size as that of the original image. An important point should be made here. Some time back, it was suggested that finding the orientation of the image is the first processing step. The technique used in the next section to find the orientation, is quite dependent on the amount of noise present in the image. Hence, the use of low pass filter is quite necessary to remove some noise points and make the orientation estimated, as close to the actual orientation as possible. In the next section, we describe a method used to find the orientation of the image.

4.3.4 Orientation

The technique used to find the orientation of the ellipse is based on what is commonly known as the principle value decomposition. A detailed description of the method can be found in Golub and Van Loan (1983, page18). The method can be described by the following steps.

1. Create an $N \times 2$ matrix A corresponding to the x & y co-ordinates of all the points that are assumed to be a part of the elliptic footprint. In the case mentioned above, all the non-zero points in the compared image are used to form the matrix.

2. Compute the centroids of the image, and subtract this value from the original image co-ordinates to form the new matrix B.

3. Find a scatter matrix of B given by Eq. 4.4

$$S = B^T B \tag{4.4}$$

The matrix S can also be referred to as the covariance matrix.

4. Compute the eigenvalue and the corresponding eigenvectors of matrix S. The matrix S is a 2×2 matrix and hence there are 2, 2×1 eigenvectors $V1, V2$.

5. The next step is to find out the higher of the two eigenvalues and the corresponding eigenvector. The eigenvector corresponding to the higher eigenvalue gives the orientation of the image. In other words, if $V1$ is the eigenvector corresponding to the higher eigenvalue, then, the ratio of $\frac{V1(1,1)}{V1(2,1)}$ indicates the slope of the major axis.

The theory behind this methodology is based on the fact that the major axis for an ellipse indicates the direction of maximum variance. The computation of the scatter matrix results in a matrix as shown in Eq. 4.5

$$S = \begin{bmatrix} \sigma_x & \sigma_{xy} \\ \sigma_{xy} & \sigma_y \end{bmatrix} \tag{4.5}$$

The scatter matrix is thus a matrix of variances and co-variances. The eigenvalues and corresponding eigenvectors of this matrix thus point to direction of maximum variance. Once the orientation of the image is found, a simple matrix that rotates the image can be used. The operations of boundary finding and the rotation can be interchanged in the present case as will be explained in the next section. It is important to note, however, that there is no morphological cleaning operations performed in this step.

An obvious requirement for a successful calculation for the orientation values is the proper registration of the image. In case the image is not recorded properly, or that it is completely broken between the extreme ends, the major axis found by the algorithm is completely opposite to the actual value. To be more precise, the algorithm performs best on the data that looks elliptic or rectangular as the case might be.

4.3.5 Boundary Finding And Rotation

Referring back to block diagram of Fig. 4.23, it can be seen that the blocks within the region marked with dotted lines can be used interchangeably. In other words, output of the system remains unaffected whether the image is rotated before its boundaries are determined or whether the boundaries are determined first before the image is rotated. The option employed in the first technique is independent of the order because there is no morphological cleaning operators employed. However, it should be noted that the second technique illustrated by the block diagram shown in Fig. 4.29 provides no such levy. In that case, the image has to be rotated before its boundaries are found as morphological operators is employed to clean the noise in the rotated image.

Going back to block diagram of Fig. 4.23, the boundary finding algorithm is applied before the rotation of the image. The boundary of the image is found in exactly the same way as that explained in Section 4.2.3. The resultant image after finding the boundary of the image depicted by Fig. 4.26 is illustrated in Fig. 4.27.

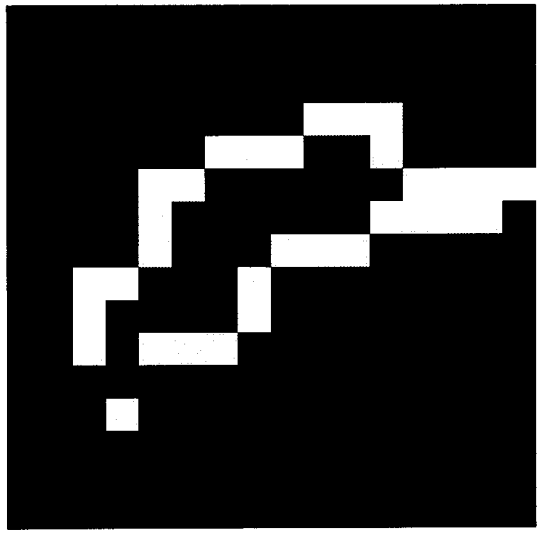


Figure 4.27: Image for RECTANGLE II after boundary finding operation

The image shown in Fig. 4.27 is now rotated with the orientation value found in Section 4.3.4. The matrix R_m used for rotating the image is presented in Eq. 4.3.5.

$$R_m = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \tag{4.6}$$

The angle θ is the angle made with the positive x-axis in the anti-clockwise direction. The image is rotated such that the major axis of the image lies along the x-axis.

The boundary image as shown in Fig. 4.27 after rotation is as shown in Fig. 4.28

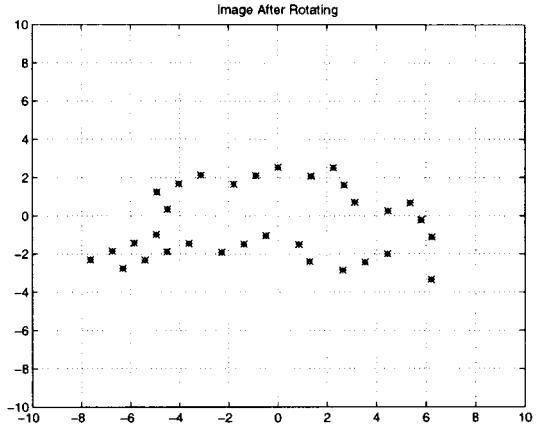


Figure 4.28: Image for RECTANGLE II after boundary finding operation

The image is obtained from a Matlab plot and hence appears to be different. An ellipse is now fit to the data. The ellipse to be fit is always centered at the origin and hence represented by Eq. 4.7.

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \tag{4.7}$$

where $2 \cdot a$ & $2 \cdot b$ are the lengths of the major and the minor axes. In the next section, we present the results of applying the technique to the data. Also, a brief description of the method used for fitting the data is suggested.

4.3.6 Least Squares Fit For ellipse

As explained earlier, it was chosen to fit an ellipse to the dataset because analytically, it is easier to describe an ellipse than a rectangle and the method of fitting the ellipse in fact follows that of a circle.

A complete derivation to compute the error function for distances between the "best" fit ellipse and a set of points is shown in Appendix D. Fitting an ellipse to a set of data requires optimization of 2 parameters, namely the lengths of the major and the minor axes. Also, a look at Eq. 4.7 suggests that the equation of the ellipse is a second order equation and hence a non-linear optimization scheme has to be employed. The optimization is performed using the MATLAB function "fmins". The function is a direct search method to find minima of an equation. It can perform non-linear optimization for up to 5 variables and is best suited for functions with a small number of local minima. The function uses simplex algorithm to perform the optimization. The results for localization algorithm applied to the test objects described in Table 4.6 are shown in Table 4.7

Table 4.7: Results for the first localization algorithm for rectangular objects

	Orientation(deg)		Length(cm)		Breadth(cm)	
	Max	Avg	Max	Avg	Max	Avg
RECTANGLE I	21	10.6	4.9	3.9	1.9	1.1
RECTANGLE II	9.0	4.8	3.4	1.9	3.4	1.7
Max=Maximum Error; Avg=Average Error						

These results, close to the size of an element (=1.3cm), are quite reasonable considering the large dimension of each element. The only large error is found to be in calculations of the length of RECTANGLE I. However, it should be noted that the calculations for the lengths and breadths are only as good as the original data. To be precise, no amount of processing can retrieve information that is not recorded in the first place. In general, it was noted that the lengths of the images recorded for RECTANGLE I were smaller compared with the actual length and the values found for the lengths are a good approximation for the recorded lengths. [1]

One of the possible sources of errors in calculating the dimensions of the ellipse is that there is essentially a significant amount of noise left in the system after the "compare-image" step. The use of morphological filtering can thus help to improve results in some cases as will be explained in the next sections.

[1] An approximate analysis of the large discrepancy between the actual and the recorded lengths was performed by manually calculating the length as recorded by the image for one of the reading. The answer found by the algorithm was found to be quite close to the manually found answer.

4.3.7 The Second Localization Algorithm for Rectangular Objects

Presence of noise can cause erroneous readings while calculating the dimensions of the rectangle. To maintain consistency with the localization of circular objects and to exploit the a priori knowledge of a rectangular footprints, it was decided to choose morphological noise cleaning. The modified block diagram for the second technique for localization of the rectangular objects is shown in Fig. 4.29. Again, the main changes compared with the block diagram depicted in Fig. 4.23 are shown by thicker blocks.

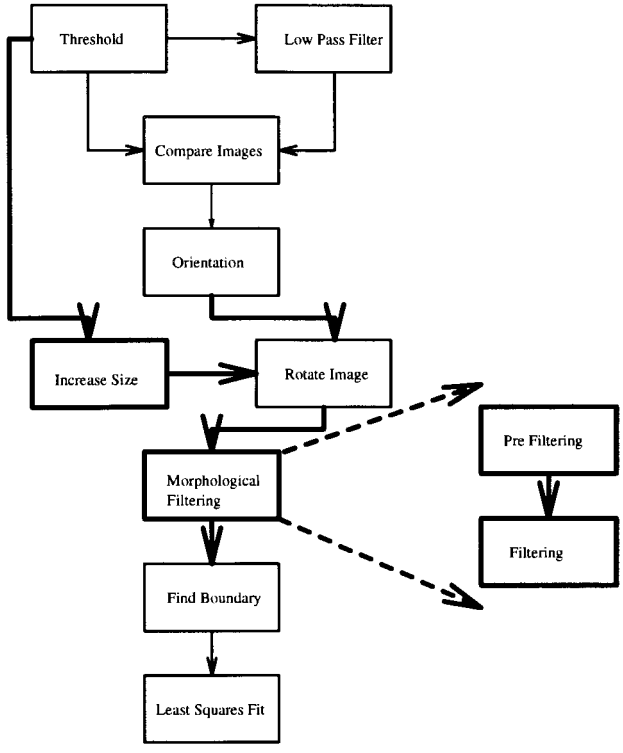


Figure 4.29: Block diagram for the second localization algorithm for rectangular objects

4.3.8 Need for Increasing Image Dimensions

It is important to recap that the size of the original image is 16×16 and that each pixel is $1\text{cm} \times 1\text{cm}$. As seen by the block diagram depicted in Fig. 4.29, morphological

filtering is applied on a rotated image. Owing to the poor resolution of the image, rotation causes a lot of rounding noise in the image. In fact, the rounding errors introduced can cause as much as a pixel shift in original image. A typical case of rounding an image is depicted in the plots shown in Fig. 4.30. The plot shows the original image in the top-left corner. The same image after subtracting the mean is shown in the top right corner. The rotated images are shown in the second row of the figure. The left bottom corner represents the rotated image before the x & y co-ordinates are rounded where as the bottom right hand corner depicts the same after the co-ordinates have been rounded. The last image on the right hand bottom corner shows that the boundary of the image is affected by the rounding operation as well as the inside of the image shows a lot of void points.

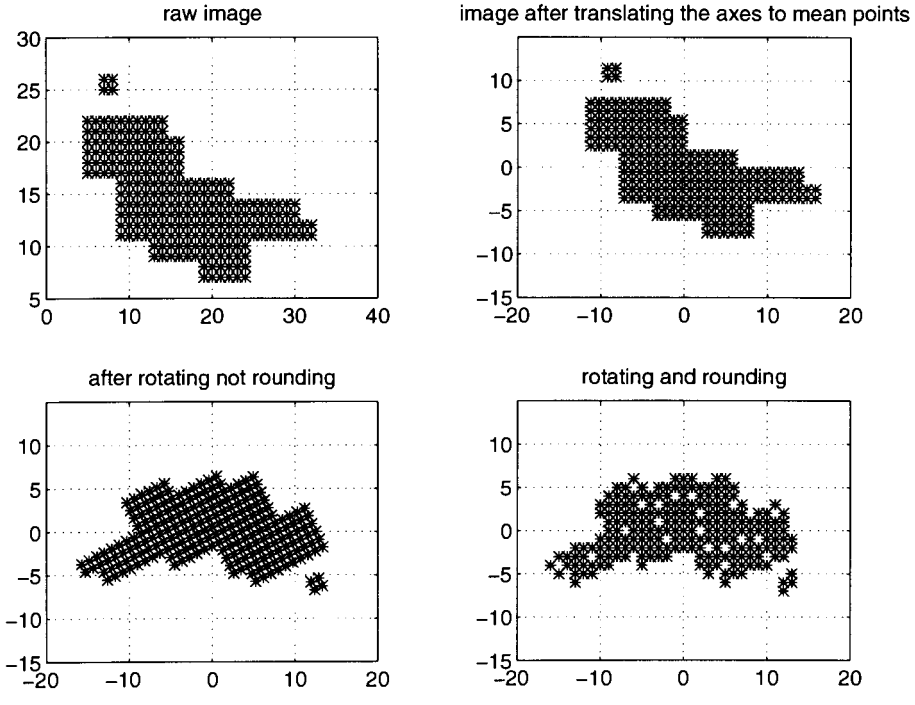


Figure 4.30: The sequence of images depicting rounding noise in the images, clock wise from top left, original image, image after subtracting the mean, image after rotating and image after rotating and rounding

The effect of rounding as mentioned before is severe because of the large dimension of each element and hence the rounding noise can be reduced by reducing the dimensions of the element. The concept explained in Section 4.2.5 can be used to minimize such errors. The size of the array was increased 10 fold to minimize the

errors due to each element to 1mm. It was decided not to reduce the size of each pixel further as the measurement itself is accurate to within ± 1mm. The image size is thus increased to 160×160 from 16×16.

Fig. 4.31 shows how a typical raw image appears after rounding operation after it undergoes an increase in size.

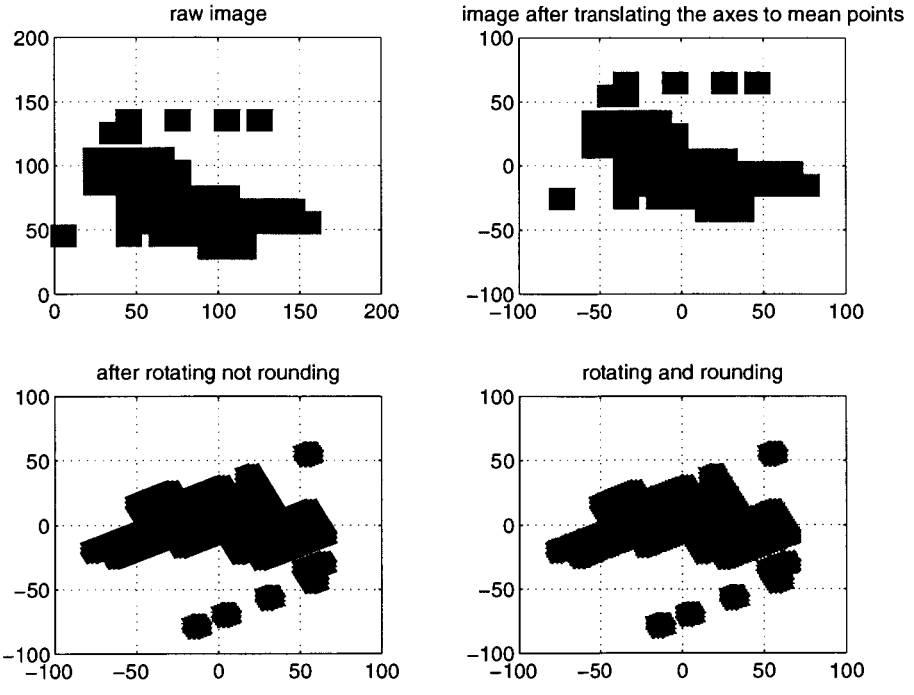


Figure 4.31: The Sequence of images depicting rounding noise in the image after increasing its size, clockwise from top left corner, original image, image after subtracting the mean, image after rotating and image after rotating and rounding

Comparing the images in the second row, it can be seen that there is little difference between the two images. The voids are still present, however, the effect of these voids is not as drastic as seen in image Fig. 4.30. The voids are dully filled using a pre-filtering stage as explained in the block diagram.

The image after rounding is ready for further processing using morphological filtering. The filtering is carried out by routines written in C and hence the images appearing later on are different from the ones presented above.

4.3.9 Morphological Filtering For Rectangular Objects

The block diagram shown in Fig. 4.29, depicts the position of the morphological filtering. It should be noted that the filtering is performed on the input image and not low pass filtered image. The morphological filtering has two blocks, namely, the pre-filtering and the filtering blocks. The functions of each of these blocks is depicted in the following sections.

Pre-filtering

The image after rounding is illustrated in Fig. 4.32.

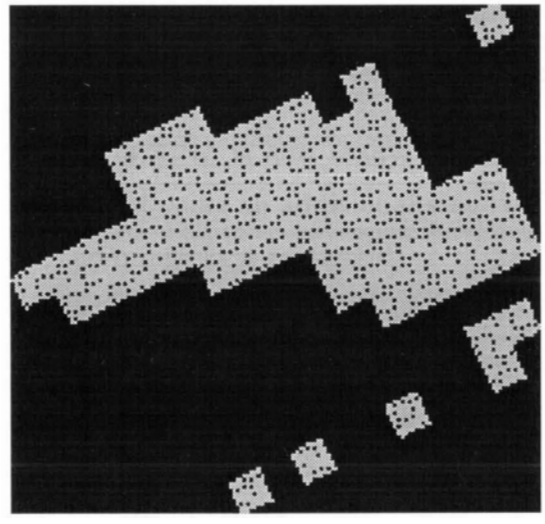
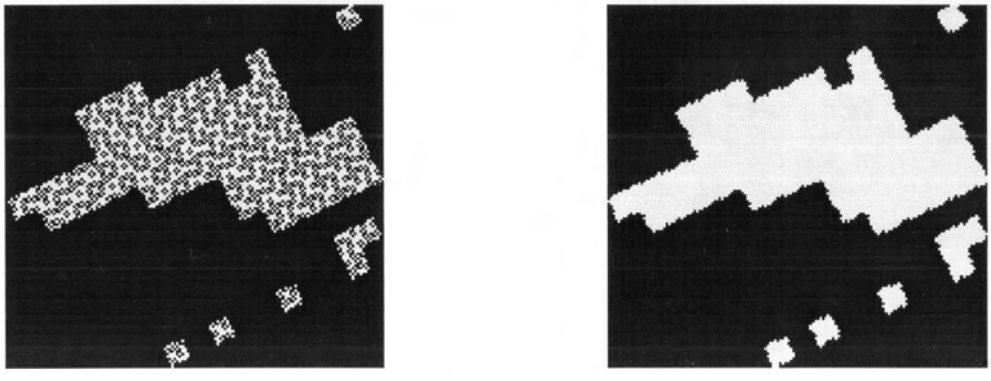


Figure 4.32: Image after rounding

The image shows a lot of void points and in some cases, a direct application of morphological filtering, even with a structuring element as small as a 3×3 in size, causes the entire image to disappear after erosion. The size of the structuring element as small as 3×3 is not very effective anyway as it does not filter the noise points. It should be remembered that a size of 3×3 is simply a pixel of size $0.3\text{mm} \times 0.3\text{mm}$ and noise pixels are as big as $1\text{cm} \times 1\text{cm}$. Hence, some operator that can fill up all the void points is necessary before we filter out the image of the noise points. A structuring element shown in Fig. 4.9(a) is used to fill up the void points. The shape and the size of the structuring element were chosen heuristically after trying a few structuring elements of varying shapes and sizes.

The operations of erosion and dilation as described in section 4.2.2 were applied and the resultant images are shown in Fig. 4.33.



(a) After erosion (b) After dilation

Figure 4.33: Effects of pre-filtering operations

The image as shown in Fig. 4.33(b) is now ready for further processing. In other words, the image can be processed using an additional stage of morphological filtering to remove noise and shape the image as a rectangle.

Filtering The Filtered Image

The structuring element used for filtering the data is different from the one used for circular objects because the shape of the image is different from a circle. The new structuring element can be seen in Fig. 4.34.

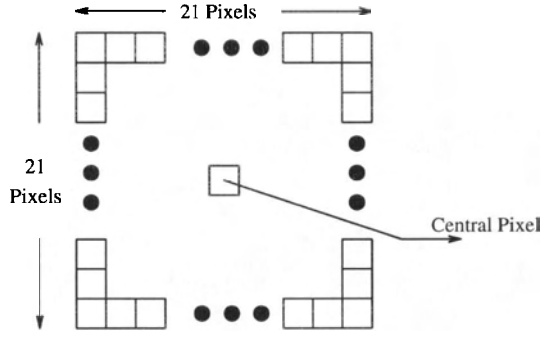
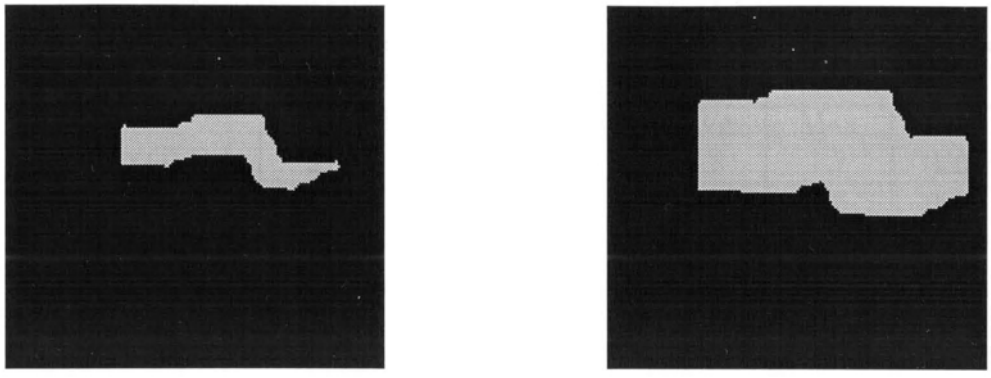


Figure 4.34: Rectangular structuring element

The size of the structuring element was heuristically chosen to be 21×21 after some experimentation. The processing is carried out in the same fashion as that described in Section 4.2.2 to generate the resultant image pairs as shown in the Fig. 4.35



(a) After erosion (b) After dilation

Figure 4.35: Effects of filtering operations

The image looks more rectangular than any of the other related images. Also, noise pixels seen in the original image are no longer present. The next step as mentioned in block diagram 4.29 is to find the boundary of the image. The method described in Section 4.2.3 is employed to find the boundary and the result is shown in Fig. 4.36. The results for the second algorithm can be seen in Table 4.8 are somewhat improved. for all parameters. The improvement is found to be more profound in case of the second object in the table.

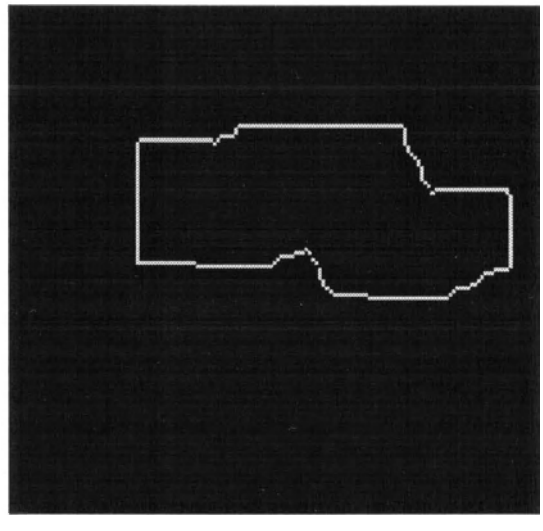


Figure 4.36: Boundary of the rectangular object

In the above sections, localization of two basic shapes have been discussed. Similar algorithms can be developed for other simple shapes. Localization routines for both the shapes assumed prior knowledge about the objects in terms of the shape of its

Table 4.8: Results for the second localization algorithm for rectangular objects

	Orientation(deg)		Length(cm)		Breadth(cm)	
	Max	Avg	Max	Avg	Max	Avg
RECTANGLE I	21	10.6	4.9	3.8	1.7	1.0
RECTANGLE II	9.0	4.8	3.3	1.6	1.6	1.3
Max=Maximum Error; Avg=Average Error						

footprint. To make a more meaningful use of the localizations routines, it is important to deduce the shapes of the objects using some other techniques. In other words, features like the weight and area may be used to index to the object shape. This forms the basis of the object identification system described in the next section. In the following section, we will describe the main motivation behind the development of an object identification system. We will discuss the operation of the system for a preliminary set of objects. Finally, some results for the techniques are presented.

4.4 Object Identification

One of the applications, often described while discussing a tactile sensor, is object identification. The sensor identifies the object based on some of its features. In fact, since the classes of objects used for identification purposes is almost always known a priori, it is quite convenient to identify the objects based on its features. Some attempts have been made to use pattern classification for tactile imaging systems as described by Chalupa, Marik, and Volf (1982), Leung and Payandeh (1994). The paper by Leung and Payandeh (1994) attempts to identify the objects placed on the FSR array using a combination of a table look up technique with some concepts from neural network. In this thesis, a feature based classification technique as shown in Christ and Sanderson (1982) is used, since the measurement patterns on the array are not very repeatable. The patterns generated for the same object vary from from time to time and measurement to measurement. Hence, it was decided to use some of the object features for classification instead of the pattern it generated. The features readily recorded by the array are the area and the weight of the object.

The localization approaches described Sections 4.2 and 4.3 use a priori knowledge about the object footprint in the algorithms. However, a feature based classification

of objects can be used to identify the object footprint by pointing to its class. If the objects with different footprints are also different in weight and area, then it might be possible to index to the object class based on these features. The block diagram shown in Fig. 4.37 illustrates this concept.

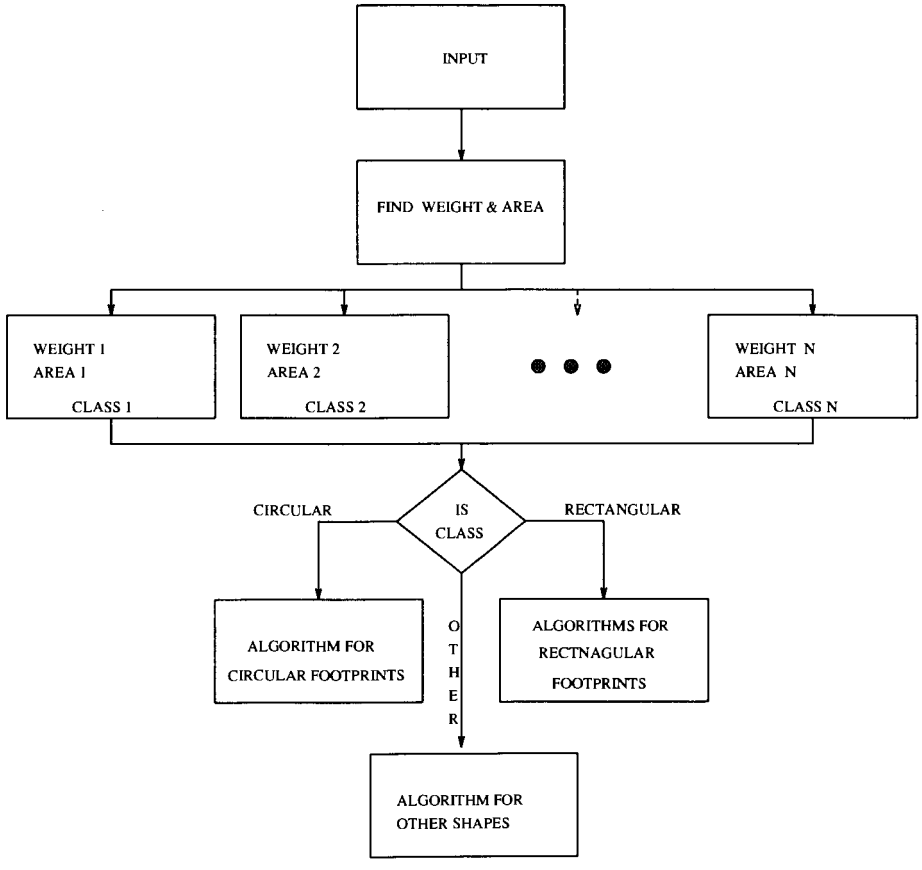


Figure 4.37: Block diagram of the indexing concept in object identification and localization system

As seen in Fig. 4.37, some combination of weight and area can be formed to separate "N" different classes of the object. As long as the object classes are easily separable, a complete identification and localization system can be developed. Also, important to note is the flexibility of the block diagram, which can permit expanding the system without any changes to the identification block.

In the following section, we describe some theory behind the classification principle, namely, the minimum distance classifier and explain the distance equation used in this thesis.

4.4.1 The Minimum Distance Classifier

The object identification system is based on the system proposed by Christ and Sanderson (1982) which uses the concept minimum distance classification to classify the object. Let W_o & A_o be the weight and the area of object o in the database and let W_{m_o}, W_{v_o}, A_{m_o} & A_{v_o} be the mean and the variance values for weight and the area respectively for object o stored in the database. The distance for object o is then calculated as shown (Christ and Sanderson 1982):

$$d_o = (\frac{W_o - W_{m_o}}{W_{v_o}})^2 + (\frac{A_o - A_{m_o}}{A_{v_o}})^2 \tag{4.8}$$

Eq. 4.8 can be modified to weigh in favor of any one of the two parameters by adding weights to each term on the right hand side of the equation. In the paper by Christ and Sanderson (1982), the first and the second order statistics, the mean and the area, are stored in a database prior to processing by presenting each object to the sensor a certain number of times. The test object is presented to the array five times and a mean of the features is used as the test weight and area. The test object is compared with every object in the database and the closest match given by the least distance is identified as the test object.

In this thesis the weight and the area of the objects are used as the features. The weight measurement is based on the curve proposed in Section 2.4.2. The mean weight is calculated for a set of 25 readings. As suggested in Section 4.1, the number 25 is a high enough to ensures a low error variance. The mean and the variance of the weight and the area are stored in the database. The formula suggested in Eq. 4.8 is modified to suite the set of readings. The term variance in Eq. 4.8 is left out. Error for each feature is however, normalized using maximum distance for that feature. The change was heuristically arrived at after some experimentation and produced the "best" results amongst all the attempted combinations. The object is identified as the one with the least distance from amongst all the distances.

Keeping the notations mentioned in Eq. 4.8 the same, the modified distance formula can be written as

$$d_o^2 = \frac{1}{max(d_{o_1})^2} \cdot d_{o_1}^2 + \frac{1}{max(d_{o_2})^2} \cdot d_{o_2}^2$$

$$where$$

$$d_{o_1}^2 = (\frac{W_o - W_{m_o}}{W_{v_o}})^2 \quad (4.9)$$

$$\&$$

$$d_{o_2}^2 = (\frac{A_o - A_{m_o}}{A_{v_o}})^2 \quad (4.10)$$

Equations 4.9 & 4.10 represent the normalizing concept discussed before. The maximum distance for feature 1 & 2 are used to normalize the corresponding distances.

In the following section, a small test is performed to evaluate the performance of the distance formula using a set of objects that are quite different from each other.

4.4.2 Initial Results

As a first step, the object identification system was tested on object that were quite different from each other. The set of object is described in Table 4.9. There are 2 classes of objects tested here, namely, the circular and the pentagonal. The latter shape was chosen to demonstrate the ability of the array to identify a variety of shapes besides just circular and rectangular.

Table 4.9: Test objects used for identification

Object	Weight(gm)	Radius(cm)
Disc I	1860	6.9
Disc II	1280	3.8
PENTAGON	475	-NA-
Disc III	6360	5.1
Disc IV	6810	7.7

25 readings of each object were tested using the distance formula mentioned in Eq. 4.9. The algorithm identified each object with 100% accuracy. It should be noted, however, that the objects selected were quite different from each other in both the areas and the weights. The test leads us to an important conclusion that the array is capable of identifying object. It is of interest to determine the performance of the array for objects with different weights but the same areas and vice versa. In the next section, we present some results that can lead us to characterize the system.

4.4.3 System Behavior

The object identification system is configured for changes in weights and the area using circular objects. The weight is found using the curves shown in Section 2.4.2. The area is found using the radius of the object calculated using circular localization algorithms presented in Section 4.2. The use of circular objects is made only for convenience, since it is easy to find the radius of the circular objects and hence the area. It should be noted that similar results apply to objects with different footprints as well. The set of test objects used in the test are described in Table 4.10.

Table 4.10: Test objects used to evaluate the system behavior

Object	Weight(gm)	Radius(cm)
Disc I	1860	6.9
Disc II	1090	6.1
Disc III	475	4.3

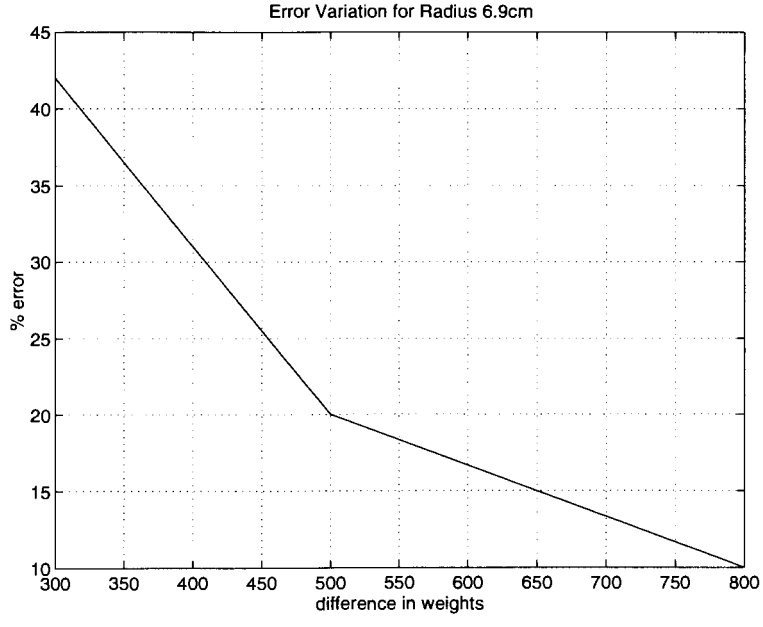


Figure 4.38: Varying weight for a base radius of 6.9cm

For each object mentioned in Table 4.10, heavier weights are simulated by adding weight on top of it. It should be noted that there is a limitation while simulating heavier weights because the readings with too many added weights are less reliable. The aim of the experiment is to evaluate the effect of variation in the weight for any

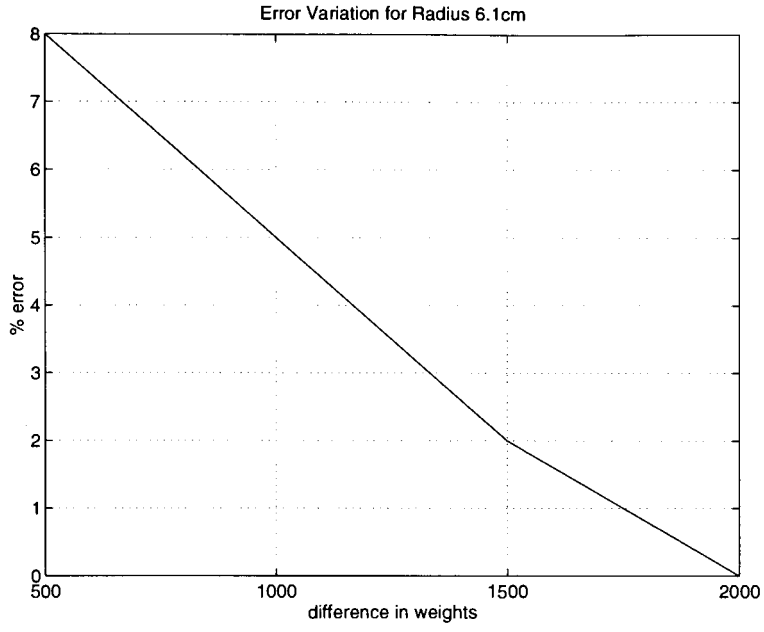


Figure 4.39: Varying weight for a base radius of 6.1cm

given radius. Therefore, at a time only two weights are considered and the errors resulting due to them are recorded. Similar results apply to more objects in the database. The graphs for DISC I, DISC II & DISC III as described in Table 4.10 are shown in Figs. 4.38, 4.39 & 4.40 respectively.

The Figs. 4.38 through to 4.40 indicate that the percentage error decreases with an increase in the difference in the weight between the two objects. The graphs for errors for radii 6.1cm & 4.3cm are almost similar. However, the graph for radius 6.9cm shows a higher error percentage. It can be argued that according to Section 2.4.3, objects with smaller radii get accurately registered compared with the larger radii for the same weight per pixel in both the cases. In all the cases, the percentage error falls to less than 5% for a difference of greater than 1kg of weight. It can be concluded that the array is capable of identifying objects with a 95% accuracy for a difference of 1000gm in weight.

Similar experiments were performed using difference in weights, but now for different radii. The curves as seen by comparing the weight variations for objects DISC I & DISC II, DISC I & DISC III and DISC II and DISC III are as shown in Figs. 4.41, 4.42 & 4.43.

As seen from Fig. 4.41 through to Fig. 4.43, the percentage error for the objects

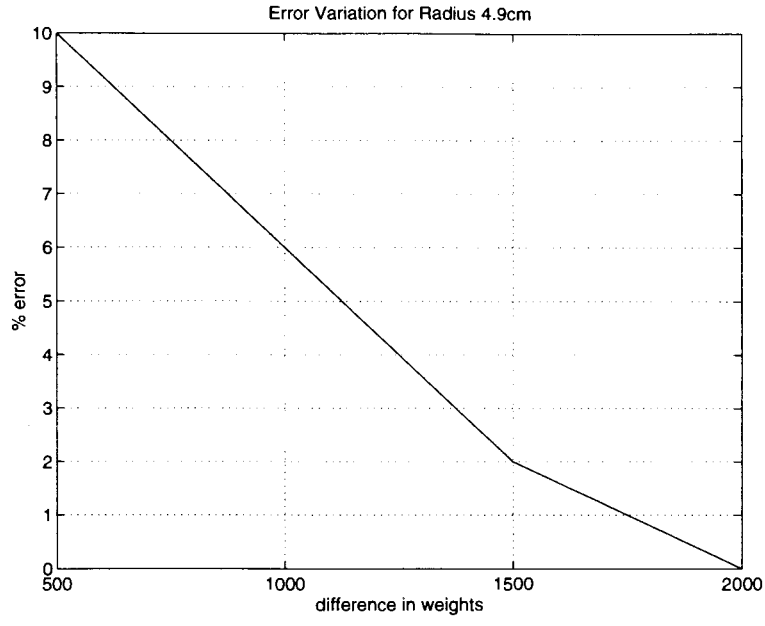


Figure 4.40: Varying weight for a base radius of 4.3cm

with the smallest difference in the radii is the most as suggested by Fig. 4.41. The performance of the other two combinations is almost similar. It should be noted that in all the three cases, the percentage error is less that 5% for a difference of greater than 1000gm in weight. It can be concluded that the percentage identification error increases for a smaller difference in the radii.

We conclude the section on object identification with an important note. Extensive testing of the system has not been performed due to the limited number of objects available and a limitation in the weight and the area of different objects used. In the following section we summarize the entire chapter.

4.5 Summary

Localization of Circular Objects Amongst all the three approaches attempted, the convex hull algorithm gave the best results. Results obtained using the convex hull approach are found to be almost accurate to within the size of one element. it should be remembered that the suggested technique is not the only technique that can be used for the localization purposes, however, it does generate fairly satisfactory results. Fine readings and greater accuracies can be

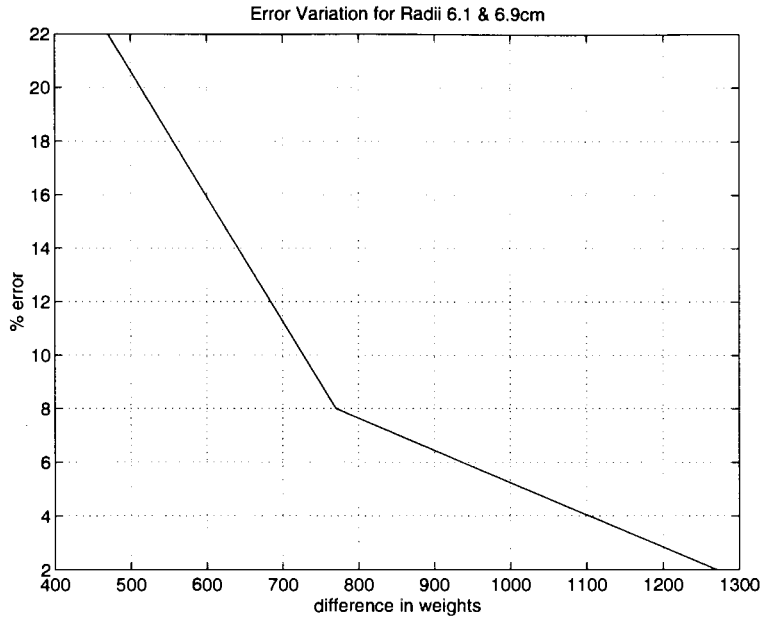


Figure 4.41: Varying weight for radii of 6.9cm & 6.1cm

expected by using smaller grids and finer resolutions which can be achieved only by physically miniaturizing the array.

Localization of Rectangular Objects The section on localizing rectangular objects can be summarized with the following remarks. The elliptic fit for rectangular objects seems to give fairly good results. Both the algorithms first find the orientation of the input image and hence the algorithms are quite sensitive to the quality of the input image. A bad footprint image results in a large error in the orientation which is propagated through to give bad final results. Morphological filtering is preceded by a pre-filtering stage that fills up voids in the image generated due to rounding of the co-ordinates. Finally, the results obtained after morphological cleaning show a dramatic improvement in cases that are severely affected by noise whereas performs almost the same as the first algorithm for other cases.

Object Identification System The object identification system presents a broad picture how it is possible to arrive at the shape of the object by classifying the objects using its features, namely the weight and the area. It was shown that the objects are identified with an accuracy of 95% if the weights are apart by

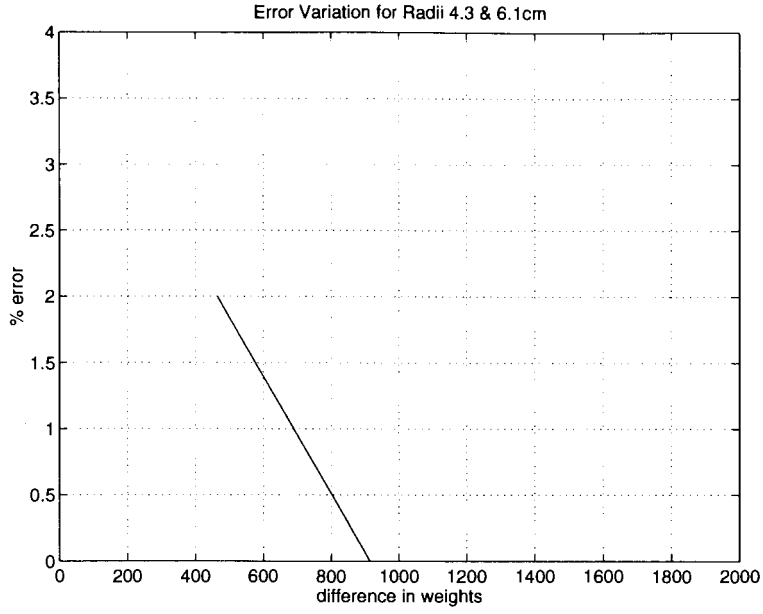


Figure 4.42: Varying weight for radii of 6.9cm & 6.1cm

1000gm and that the percentage error goes on increasing for the same weight difference as the area of the objects are made more similar.

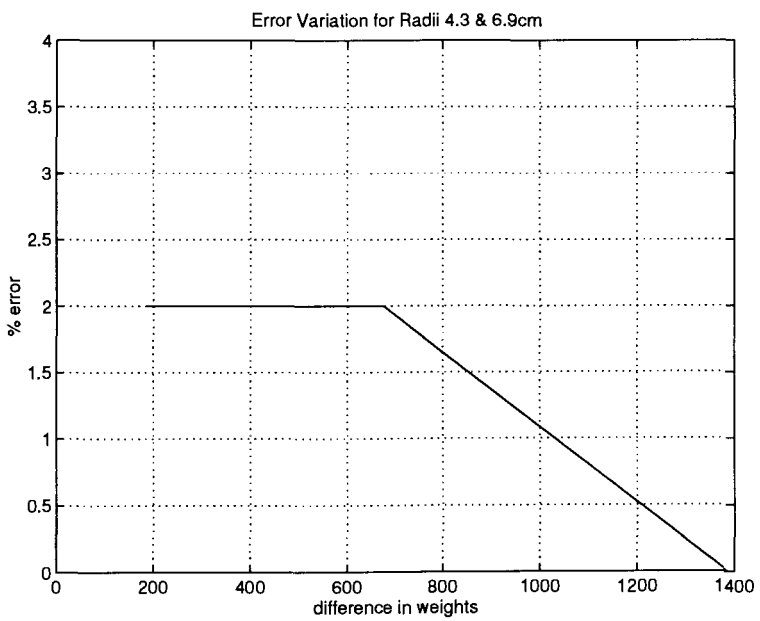


Figure 4.43: Varying weight for radii of 6.1cm & 4.3cm

Chapter 5

Bayesian Image Restoration Techniques

In Chapters 3 and 4, we described the algorithms used for extracting features and locating the objects. The techniques described were heuristic in nature. In this chapter, we suggest a model based technique for extracting similar information from the object. The following sections present some theoretical background to the technique as well as some simulation results.

5.1 Bayesian Techniques: An Overview

In the previous chapters, we presented some images and showed how "noise" affected the readability of the images. In many applications, it is necessary that the pressure readings be reconstructed from the noisy data as accurately as possible. Bayesian techniques are commonly employed to recover data from the noisy images and hence are applicable to our problem as well.

One of the first results in estimating and interpreting the results on tactile data using Bayesian techniques was published by Nowlin (1991). In this paper, a comparison is made between the results obtained using least squares and the Bayesian techniques. The paper suggests that in the presence of Gaussian noise, the latter technique works much better and provides better results. If y is the noisy measurement of the real data x, then according to Bayes theorem,

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} \tag{5.1}$$

The paper suggests maximizing the left hand side of Eq. 5.1 with respect to x. However, the denominator of the right hand side of Eq. 5.1 is independent of x. Hence maximizing $p(y|x)p(x)$ maximizes Eq. 5.1. This means that prior distributions on x and $y|x$ need to be known. In the problem tackled by Nowlin (1991), these distributions are assumed to be normal with zero means. However, in general, it is quite difficult to find the prior distributions on x and $y|x$, especially the latter. Besides, another limitation of the suggested technique is that it works only for those distributions for which the product $p(x)p(y|x)$ is unimodal. This might not always be the case.

The technique used in this thesis for image restoration is based on the work by Geman and Geman (1984) and Hargreaves (1994).

5.1.1 Simple Overview of the Technique

Geman and Geman suggest exploiting the relationship between the neighboring pixels to arrive at the most likely posterior distribution given corrupted input data. The paper also suggests a technique that maximizes this distribution.

The paper treats the image field to be a Markov Random Field (MRF). The definition of Markov random field means that the value at any one pixel is dependent only on the pixels in its neighborhood. In other words, the first step in formulating an MRF involves defining a relationship between pixels that determines the neighbors for any one pixel. Such a relationship results in a neighborhood system, which is formally defined in Section 5.1.2. Appropriate energy function can be used to enforce smoothness constraint in the image.

The set of possible values at any one pixel is defined as the state space associated with that pixel. For simplicity, the state space is assumed to be the same for all the pixels in the image. The neighborhood system simply indicates the set of pixels that affect the value at a given pixel, but does not provide the actual relation between these pixels. Geman and Geman suggest that the pixels in the neighborhood are related to each other by a Gibbs distribution. In fact, according to Hammersely-Clifford theorem presented by Geman and Geman, the image field is an MRF if and

only if the neighborhood pixels follow a Gibbs distribution. The Gibbs distribution provides a set of mathematical constraints between the pixels in the form of potential and energy functions; which are formally defined in Section 5.1.2. What this means is that each configuration of pixels will be associated with some energy. It turns out that computing the maximum a posteriori estimate (MAP) estimate is equivalent to minimizing this energy associated function over the image.

It is important to note that there is a large number of total possible image configurations. In other words, for a state space of L and for an image of size $M \times N$, the total number of possible images is LMN. A direct computation of the global minimum of the energy function is thus quite cumbersome and hence a sampling technique is employed. Geman and Geman suggest using simulated annealing and the "Gibbs sampler" to minimize the energy. Section 5.2.2 formally defines both the Gibbs sampler and the simulated annealing.

The next section describes the mathematical model behind each of the formulations explained above.

5.1.2 Gibbs Distribution

In this section, we present the definition of Gibbs distribution and explain each of the associated terms with the definition.

Gibbs Distribution

Consider an $m \times m$ lattice defined in Eq. 5.2

$$Z_m = (i, j) : 1 \leq i, j \leq m \tag{5.2}$$

and let the set of pixels (sites) in Z_m be represented by $S = \{s_1, s_2, s_3, \ldots, s_N\}$ where $N = m^2$. Let a family of random variables indexed by S be denoted by X such that $X = \{X_s, s \in S\}$. As mentioned in Section 5.1.1, the simplest case is to assume the same state space, $\wedge$, for all the variables X_s such that $\wedge = \{0, 1, 2, \ldots, L-1\}$ and $X_s \in \wedge$ for all s. L denotes the total number of levels in the state space.

Let Ω be the set of all possible configurations given by Eq. 5.3.

$$\Omega = \{\omega = (x_{s_1}, \ldots, x_{s_N}) : x_{s_i} \in \wedge, 1 \leq i \leq N\} \tag{5.3}$$

where the occurrence of any event in the random field, namely, $\{X_{s_1} = x_{s_1}, \ldots, X_{S_N} = x_{s_N}\}$ is abbreviated as ω.

A Gibbs distribution over $\{S, \mathcal{G}\}$, where $\mathcal{G}$ is a neighborhood system defined in the next section, is a probability measure π on S defined as given by Eq. 5.4.

$$\pi(\omega) = \frac{1}{Z} e^{-\frac{U(\omega)}{T}} \tag{5.4}$$

where $U(\omega)$, *energy function*, is defined as:

$$U(\omega) = \sum_{c \in \mathcal{C}} V_c(\omega) \tag{5.5}$$

The term $V_c(\omega)$ is defined as the *potential function* and is dependent on only those samples of X which belong to the set of "cliques" $\mathcal{C}$ defined in the following section. A sample energy function is also defined in Eq. 5.7. The term Z in Eq. 5.4, termed as the partition function, normalizes Eq. 5.4 so that $\pi(\omega)$ is a valid probability density function. Finally, the term T used in Eq. 5.4 is a constant and is analogous to the temperature term in the physical process.

The following sections explains the main concepts behind the terms associated with the Gibbs distribution, namely, the *neighborhood system and the cliques.*

Neighborhood System

Every pixel is associated with a set of pixels surrounding it. Some constraints define the neighborhood system for any such pixel.

The neighborhood system $\mathcal{G}$ on S mentioned in Eq. 5.4, can be defined as $\mathcal{G} = \{\mathcal{G}_s, s \in S\}$ with the following constraints:

1. $s \notin \mathcal{G}_s$ and
2. $s \in \mathcal{G}_r \Leftrightarrow r \in \mathcal{G}_s$

This means that a point does not belong to its own neighborhood system and that if s in the neighborhood of r, then r is in the neighborhood of s. Images are 2-dimensional and hence the pixels are addressed by their coordinates (m, n). Fig. 5.1 shows the neighborhood systems for orders 1 to 3, where $\mathcal{G}^k$ denotes a k^{th} order system. It is important to note that the higher order neighborhoods contain all the

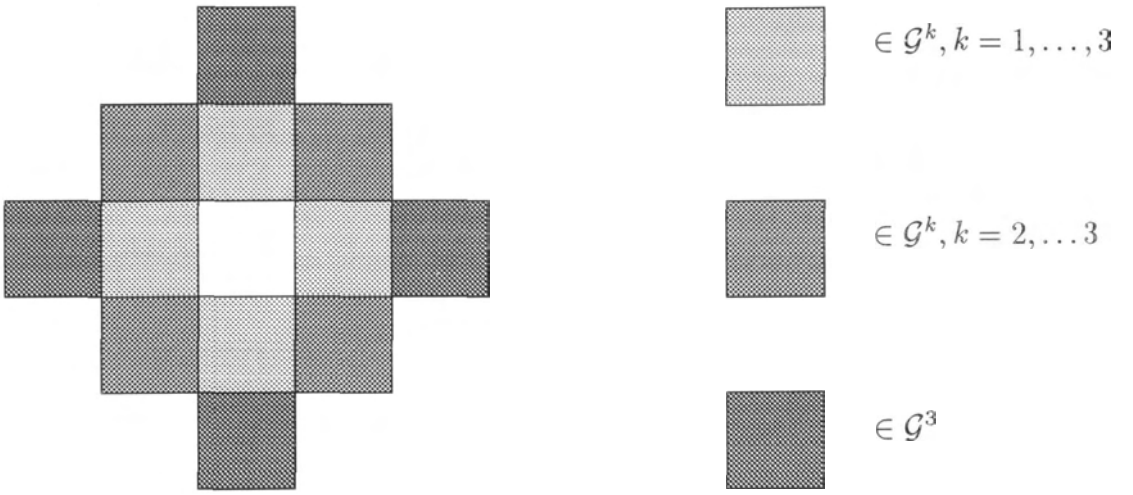


Figure 5.1: Neighborhoods for order 1, 2 & 3 (Hargreaves, 1994)

pixels of lower order neighborhoods. As shown in Fig. 5.1, the pixels marked by $\mathcal{G}^k, k = 1, \ldots, 3$, belong to all of the three neighborhood systems.

The figure shows that the first order neighborhood consists of pixels located at $(m-1, n), (m, n-1), (m+1, n)$ & $(m, n+1)$ with respect to the central pixel which is at (m, n). The total number of pixels in the neighborhood increases with the order of the neighborhood system. The site $(m+2, n)$ does not belong to the second order neighborhood system of the pixel (m, n) and hence, according to condition 2 mentioned above, pixel (m, n) does not belong to the second order neighborhood of pixel $(m+2, n)$.

In the thesis, only the first order neighborhood system has been used because higher order system increase the computational complexity significantly while providing only marginal performance improvements (Hargreaves 1994).

The set C, used in Eq. 5.4, namely, the set of cliques, is dependent on the neighborhood system and is defined in the following section.

Cliques

Associated with the neighborhood systems, is a set of cliques. A subset $c \subseteq S$, is a clique if every pair of distinct sites in c are neighbors. Let the set of cliques be defined as $\mathcal{C}$. In mathematical terms, the conditions can be stated as follows (Hargreaves 1994):

1. Two pixels in a clique, c, are neighbors. In other words, if (m,n),$(p,q) \in c$, then $(m,n) \neq (p,q) \Rightarrow (m,n) \in \mathcal{G}_{(p,q)}$.
2. c can consist of a single pixel.

The set of cliques associated with the first order neighborhoods is shown in Figs. 5.2.

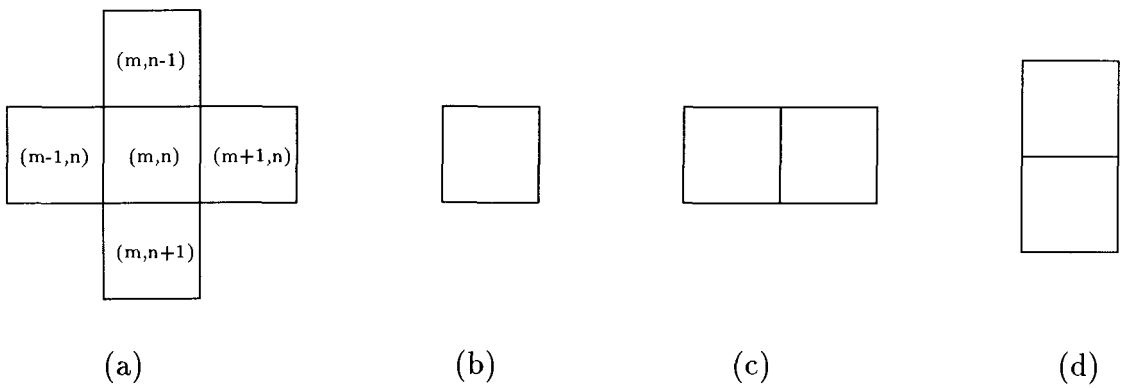


Figure 5.2: First order neighborhood system (a), associated cliques: (b)single element (c) & (d)two element cliques

The figure shows the set of cliques over which the energy function defined in Eq. 5.4 is evaluated.

Energy Function

A sample energy function can be defined as

$$U(\omega) = \sum_{c \in \mathcal{C}} (V_{(m,n)} - V_{(m_c,n_c)})^2 \tag{5.6}$$

and

$$en(\omega) = U(\omega) + \sum_{m,n} (in(m,n) - it(m,n))^2 \tag{5.7}$$

where $V_{(m,n)}$ is the center pixel, $V_{(m_c,n_c)}$ defines the location of the clique element, $in(m,n)$ defines the input image and $it(m,n)$ defines the current image. Eq. 5.6 defines a sample potential function, which enforces a smoothness constraint by summing over the difference in the values between the pixels forming the cliques. Eq. 5.7 defines a sample energy function that sums the potential function with an additional term that is evaluated over the entire image. This term evaluates the difference between the input image and the current image at all pixels.

5.1.3 Markov Random Fields

As mentioned in Section 5.1.1, the image restoration algorithm described in Geman and Geman (1984) assumes the image field to be a Markov random field. A Markov random field can be described as having the following properties:

1. $P(X = \omega) \geq 0$ for all $\omega \in \Omega$.
2. and the local characteristics defined by:

$$P(X_s = x_s | X_r = x_r, r \neq s) = P(X_s = x_s | X_r = x_r, r \in \mathcal{G}_s) \tag{5.8}$$

where P defines the probability measure, X_s & X_r are any occurrence of the random fields as mentioned in Eq. 5.3. The above conditions simply mean that the probability of having a particular value at a pixel is dependent only on those sites that belong to the neighborhood of the pixel.

Ordinarily, it is quite difficult to determine the characteristics of the MRF directly. However, as mentioned in Section 5.1.1, according to Hammersley-Clifford theorem, X is an MRF with respect to a neighborhood $\mathcal{G}$ *iff* the probability distribution of its sample realizations is a Gibbs distribution as given by Eq. 5.4 with respect to $\mathcal{G}$. Due to the theorem, local characteristics of the MRF do not have to be determined while performing the necessary analysis on the image. The probability of a possible image configuration, given by ω, is thus a function of the potential function and in turn the energy function, given by Eq. 5.5.

The above theorem leads us to combine Eq. 5.4 and 5.8 as:

$$P(X_s = x_s | X_r = x_r, r \neq s) = \frac{1}{Z} e^{-\frac{U(\omega)}{T}} \tag{5.9}$$

where

$$Z = \sum_{x \in \Lambda} e^{-\frac{1}{T} \sum_{c: s \in C} V_c(\omega^x)} \tag{5.10}$$

The above equations suggest that the probability of a value at a pixel indexed by X_s depends on those pixels that belong to the cliques of the neighborhood. This means that once the neighborhood system is defined, evaluating the conditional probability term is quite straightforward, unlike formulations explained in Nowlin (1991).

5.2 MAP Estimation

The purpose of the estimation is to determine the posterior distribution of the original image given data "g". In mathematical terms, we need to find, $P(X = \omega|G = g)$. Using Bayes theorem,

$$P(X = \omega|G = g) = \frac{P(G = g|X = \omega)P(X = \omega)}{P(G = g)} \tag{5.11}$$

Finding the MAP estimate means maximizing the term $P(G = g|X = \omega)P(X = \omega)$. The term $P(G = g)$ is a constant whereas the term $P(X = \omega)$ is given by Eq. 5.9. This leaves only one unknown term in the equation and that is $P(G = g|X = \omega)$ which can be evaluated using the model as described in the next section.

Observation Model

This model takes into account both the distortion and the noise that affects the original image, F. Let H be a blurring matrix and N be a noise process, then the corrupted image can be represented as $\psi(\phi(H(F)), N)$ where ψ and ϕ are some functions. This is the recorded image G. Mathematically,

$$G = \phi(H(F)) \odot N \tag{5.12}$$

where $\odot$ represent some "invertible" process.

The noise in our system is assumed to be additive Gaussian noise for simplicity. More work needs to be done to come up with a more accurate approximation of the noise process. The noise in our system can thus be simply written as

$$G = F + N \tag{5.13}$$

5.2.1 MAP estimate

Consider the term $P(G = g|X = \omega)$. From Eq. 5.9, and Eq. 5.13, we can write,

$$\begin{aligned} P(G = g|X = \omega) &= P((F + N) = g|X = \omega) \\ &= P(N = \Phi(F, G)|X = \omega) \end{aligned}$$

$$= P(N = \Phi(F, G)) \tag{5.14}$$

Eq. 5.14 requires that the noise be invertible and independent of X; within this constraints, any type of noise to be added to the system. Combining Equations 5.4, 5.14 & 5.11, we can express the maximum a posteriori probability estimate as

$$\begin{aligned} P(X = \omega | G = g) &= \frac{P(N = \Phi(F, G))P(X = \omega)}{P(G = g)} \\ &= \frac{1}{Z} e^{-\frac{U(\omega)}{T}} \cdot \frac{P(N = \Phi(F, G))}{P(G = g)} \\ &= \frac{e^{-\frac{U^P(\omega)}{T}}}{Z^p} \end{aligned} \tag{5.15}$$

$$\tag{5.16}$$

where Z^P is a normalizing constant as before and $U^P(\omega)$ notes the a posterior energy function. It is shown by Geman and Geman that in case of additive white Gaussian noise,

$$U^P(\omega) = U(\omega) + \| F - N \|^2 \tag{5.17}$$

The second term is the sum of the squared difference of every pixel as used in Eq. 5.7. However, in reality, the noise in our system is likely not Gaussian or even additive in nature and hence the evaluation of the second term needs some attention. In the thesis, the energy function is evaluated using Eq. 5.17 as shown in Eq. 5.7 and the changes are left for the future studies.

5.2.2 Stochastic Solution to MAP Estimate

In order to find a dependable restored image, it is necessary to find the MAP estimate with accuracy. The thesis uses the MAP estimation technique employed by Geman and Geman which is stochastic in nature and uses what is known as *simulated annealing*. Eq. 5.15, maximizes the a posteriori distribution by maximizing the right hand side of the equation. The right hand side of the expression consists of $\frac{e^{-U(\omega)}}{T}$ and hence we need to minimize the total energy associated with the system as given by $U(\omega)$.

5.2.3 Gibbs Sampler

Simulated annealing is similar in nature to the physical annealing process carried out for cooling of steels or metals. However, in order for simulated annealing to be computationally realizable, it is necessary to generate samples from the a posteriori estimate. One technique for doing this is called as Gibbs sampler

Let the configuration for the image at time t be given by

$$X(t) = (X_{s_1}(t), X_{s_2}(t), \ldots, X_{s_N}(t)) \tag{5.18}$$

where the state of site s is a random variable $X_s(t)$ with the values in the state space set $\wedge$. The starting configuration $X(0)$ is random. Each lattice point in Eq. 5.2 is referred to as a site.

At every step only "one" site can undergo a possible change which means that $X(t-1)$ and $X(t)$ differ in at most one co-ordinate. Let, $n_1, n_2, \ldots, n_t, \ldots$ is the sequence in which the sites are visited for a possible change. This means that $n_t \in S$ and $X_{s_i}(t) = X_{s_i}(t-1)$ if $i \neq n_t$. The possible change is made by drawing a sample from the local characteristics of π for $s = n_t$ and $\omega = X(t-1)$. Mathematically, the condition that describes the possible change can be written as

$$P(X_s(t) = x_s, s \in S) = \pi(X_s(t) = x_s | X_s(t-1) = x_s, s \neq n_t) P(X_s(t-1) = x_s, s \neq n_t) \tag{5.19}$$

Eq. 5.19 means that the change is made based on the sample generated from the local conditional probability distribution of X_{n_t}. The block diagram of Gibbs sampler can be as shown in Fig. 5.3.

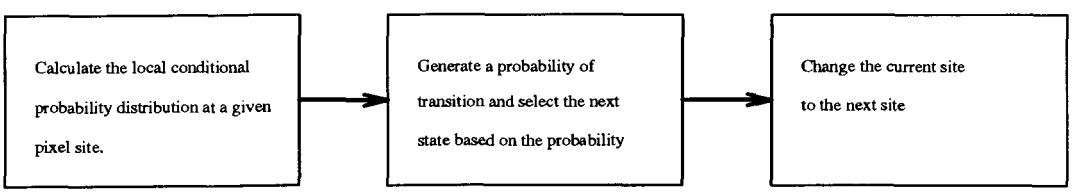


Figure 5.3: Block diagram of the Gibbs sampler algorithm

Actual Algorithm

The step by step description of the algorithm is as follows:

1. For each site, calculate the energy associated with each state of the total state space at that site.
2. Normalize the energies such that they fall within the boundaries of the maximum and the minimum floating point capacity of the computer.
3. Calculate the probability of transition for each of the state using Eq. 5.19.
4. Generate a random number on an interval $[0, 1]$ and select the next state based on the value obtained.

Simulated Annealing

As explained above, simulated annealing is employed to generate states at any given pixel. A direct search for a global minima is computationally expensive. For example, for L levels in the state space and an $M \times N$ sized image, the total number of images is L^{MN} and the total MNL number of states is MNL for each image. Even for a small M, N, L these values become quite large and a direct search for a global minima is computationally not viable. In our case, the value of L is chosen to be 200.

The technique used for simulated annealing in this thesis is adopted from Hargreaves (1994) and the flow chart for the technique can be seen in Fig. 5.4.

The algorithm is based on the cooling of solids to attain a state of equilibrium and a minimum energy. The algorithm lowers the temperature according to an *annealing schedule*. The starting value of the temperature, annealing schedule, total number of iterations and state space for each pixels are design parameters. The typical values for starting temperature in pour application is 5.0 and the annealing schedule is described by $T_{n+1} = 0.9T_n$. It is important to note that too low a starting temperature causes the algorithm to be stuck in some local minimas whereas too high a value causes a long time to reach the global minima. The annealing schedule is also quite important as large changes in temperature values might cause the system to freeze up in some local minimum and smaller changes mean that the time required to reach the equilibrium state will be excessive.

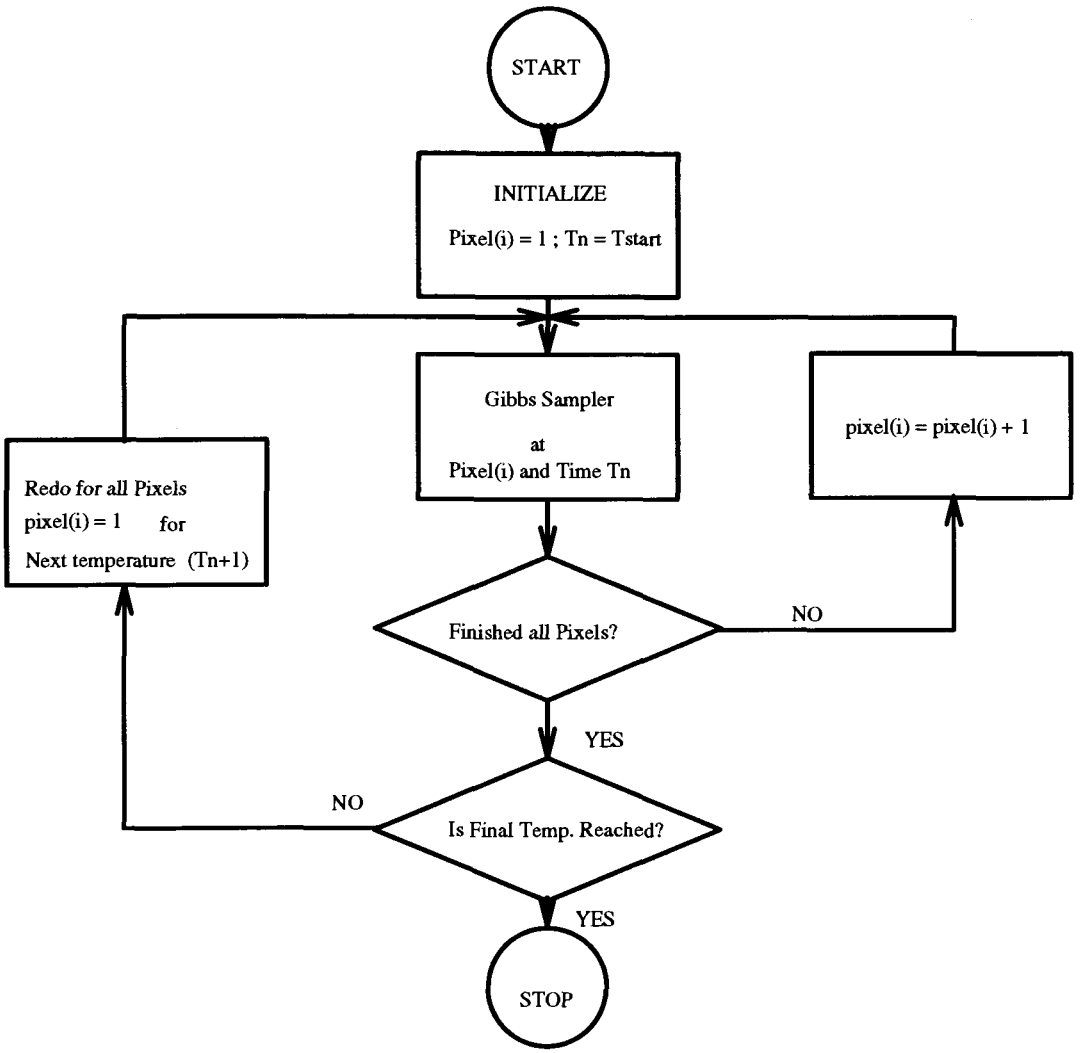


Figure 5.4: Block diagram of the simulated annealing algorithm

5.3 Discontinuities in Images

The above sections describe image models that assume that all the pixels in an image belong to one smooth image. As shown in Eq. 5.6 indicates how a smoothness constraint is enforced. No discontinuities in the form of edges and boundaries are incorporated in this equation. While, the model is useful to smooth out noise spikes and generate a uniform image, its performance is similar to that of a low pass filter. An enhancement has been suggested by Geman and Geman (1984), which was adopted by Konrad (1989). The thesis uses the algorithm employed by Hargreaves (1994).

5.3.1 Energy Function With Line Field

The incorporation of the line field permits discontinuities in the otherwise smooth image model as described above. In other words, the energy associated between the pixels on the boundary of the object may be significantly different than those belonging to the interior of the object in order to prevent smoothing across the object boundaries. The energy function is thus modified as shown in Eq. 5.20. As seen from the Eq. 5.20, if the line field between the pixels is ON, in other words, there is a likely edge between the two pixels, then the contribution to the total energy is zero.

$$U(\omega, l) = \sum_{c=\{x_i,x_j\}\in C} V(\omega, c) \cdot [1 - lf < x_i, x_j >] \tag{5.20}$$

where $lf < x_i, x_j >$ represents the line field pixel between the two site pixels x_i and x_j. The presence of line field causes a break in the image and as such the potential contribution due to that clique is equal to zero. The line field energies explained later are chosen in such a way that isolated single edges are punished and a smooth line field is encouraged.

5.3.2 Line Field Model

As in a continuous smooth modeling explained above, the main design term affecting the performance is the energy associated with the line fields. The total energy associated with any line field is given by Eq. 5.21 [1]

$$U_l(\omega, l) = \lambda_d \cdot \sum_{l<x_i,x_j>} \| x_i - x_j \|^2 + \lambda_l \cdot \sum_{c_l : l<x_i,x_j>\in C_l} V_l(\omega, c_l) \tag{5.21}$$

where $l < x_i, x_j >$ defines the line field between two pixels and c_l and C_l defines the line field cliques and the set of line field cliques respectively. λ_l and λ_d are some constants. Finally V_l is the line field potential.

Eq. 5.21 is made of two factors. The contribution due to the difference between the pixel sites at $< x_i, x_j >$ and the potential due to the line fields cliques centered around line field pixel between $< x_i, x_j >$. The parameters λ_d and λ_g allow a possible balance between the contributions made by the energies at sites $< x_i, x_j >$ and the

[1] The equation is adopted from the one used by Hargreaves (1994)

line field. These are design parameters and are adjusted based on the expected values at the sites.

Neighborhoods and Cliques for Line Fields

As before, the neighborhoods and the cliques have to be defined before computing the energies due to the line field. Fig. 5.5 describes the first order neighborhood and the presence of line field pixels between them.

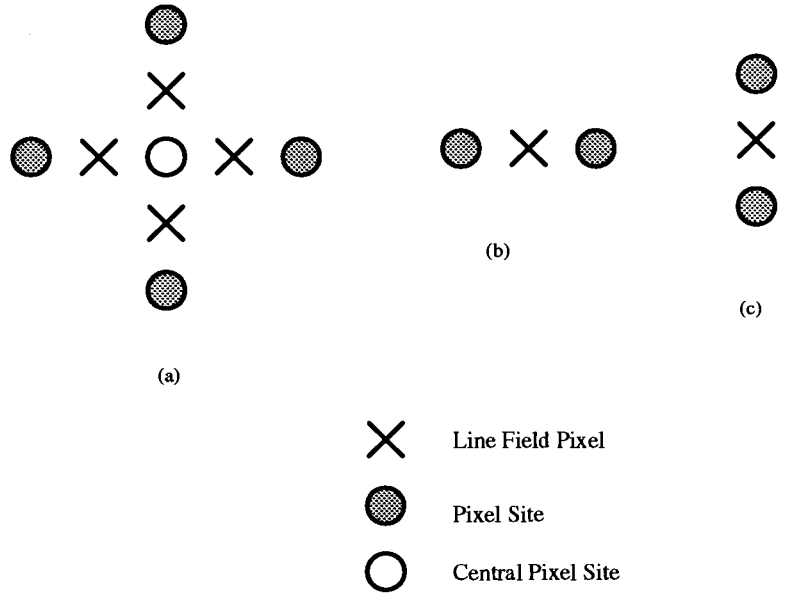


Figure 5.5: (a)First order neighborhood system with the line field pixels, (b)horizontal and (c)vertical cliques

The figure explains the concept of the line field and shows the associated line field pixels. The cliques associated with the line field can be seen in Fig. 5.6

Besides the cliques shown in Fig. 5.6, there exists single element cliques corresponding to each line site. The contribution of each single element cliques is dependent on whether the clique is "ON" or "OFF".

The total energy contributed by the second term of Eq. 5.21 can be written as

$$V_l(\omega, c_l) = V_{l_4}(\omega, c_l) + V_{l_2}(\omega, c_l) + V_{l_1}(\omega, c_l) \tag{5.22}$$

where the subscript l denotes the potential due to 4,2 or 1 element cliques.

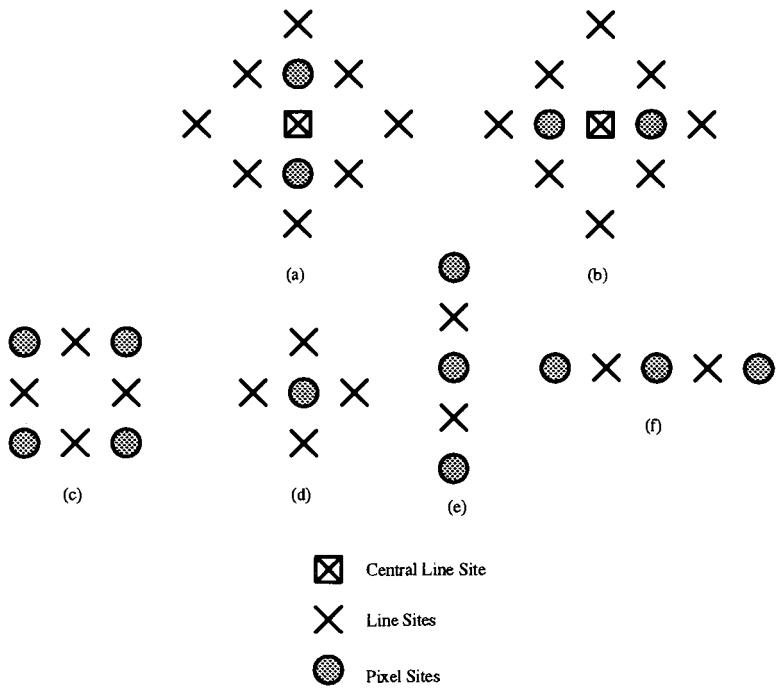


Figure 5.6: (a)Horizontal & (b)vertical line fields with (c),(d)four element & (e),(f)two element cliques

Finally, energies associated with the presence of each pattern of line field have been given in Konrad (1989) and used by Hargreaves (1994). The calculation of the energies penalizes the presence of single isolated line sites and encourages smooth continuous line field model. Fig. 5.7 indicates the line field energies used in this thesis.

5.4 Simulation Results

There are many parameters that need to be chosen for this technique to work well. The test image was chosen as shown in the top left hand corner of Fig. 5.8 because it represents a uniform distribution over the array, which is similar to the one expected when an ideal array is loaded by an object. The image was corrupted by the addition of Gaussian noise because the noise model presented in Section 5.2, uses Gaussian noise. The corrupted image as shown on the top right hand corner of images illustrated in Fig. 5.8 is subjected to different processing conditions to determine the effects of the following:

1. The temperature parameter and the annealing schedule,

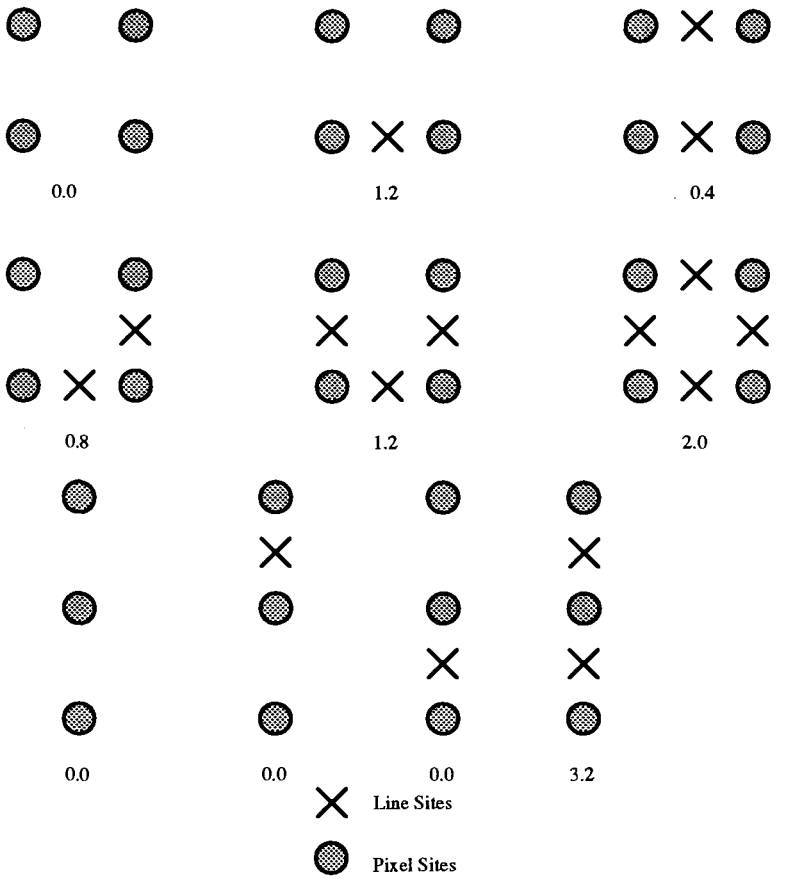


Figure 5.7: Energy associated with different patterns of line fields (Hargreaves(1994))

2. The number of iterations needed,
3. Values of the state space for an element,
4. The presence of the line field,
5. Parameters λ_d and λ_d used in Eq. 5.21.

5.4.1 Test Results

The simulation was performed with heuristically chosen starting temperature values of 5.0 and 2.0. The values were chosen to ensure a sufficiently high temperature values to prevent getting stuck in local minimas to begin with. The effect of the starting temperature was found to be negligible as the final images appeared to be the same. The annealing schedule was made to vary from $T_{n+1} = 0.9 \cdot T_n$ to $T_{n+1} = 0.95 \cdot T_n$ but did not affect the results much. The latter one produced a more gradual change in

the images. The temperature parameter was thus chosen to be 5.0 and the annealing schedule $T_{n+1} = 0.9 \cdot T_n$ for all further experiments. The total number of iterations were heuristically chosen to be 110 because the images did not change much after a value of about 95 iterations. The state space value was tested in steps of 50 from 50 to 250. The images at lower values of state space (50 to 150) showed clearly visible steps. However a value of 200 resulted in a fairly smooth image. There was little or no difference between the images by varying the state space from 200 to 250 but resulted in longer time of execution and hence the state space was chosen to be 200. Fig. 5.8 illustrates the filtering operation.

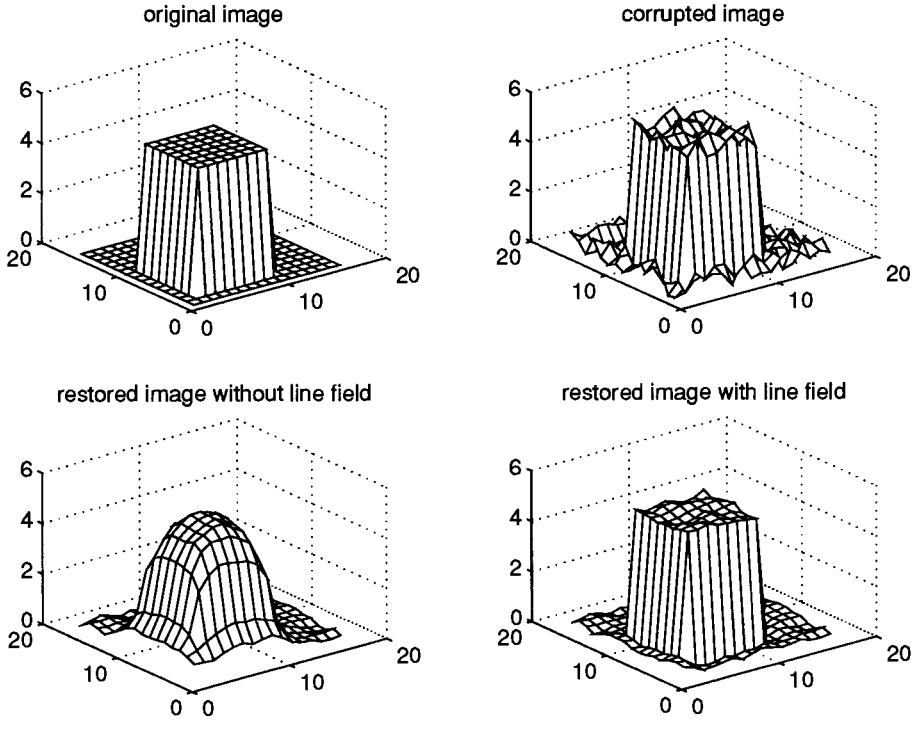


Figure 5.8: (a)Original image, (b)Corrupted image (c)Image without line field (d)Image after line field

An interesting observation can be made from the bottom row of the images in Fig. 5.8. The images illustrate the effect of implementing the line field. It was mentioned before, that the algorithm without the line field behaves simply as a low pass filter and tends to smooth the image as seen in left hand corner image in Fig. 5.8. However, the line field implantation shows that the image is no longer constraint to be smooth throughout. In fact, the original image and the restored image seen in

Fig. 5.8 look quite similar.

An image that simply represents the line field can be seen in Fig. 5.9.

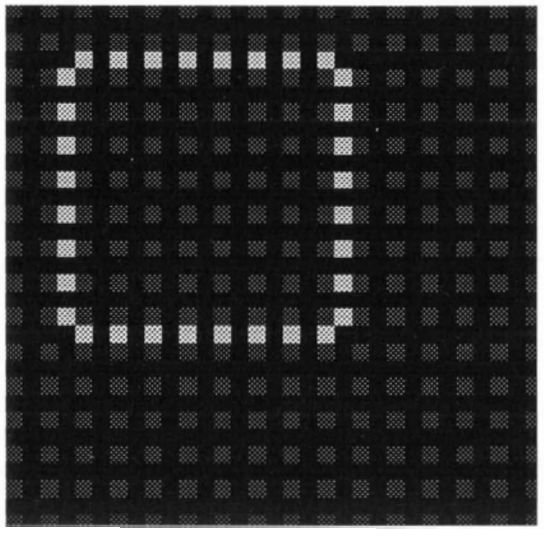


Figure 5.9: Line field plot

The void spaces are a result of the way the line field is displayed. The dimensions of the rectangle as shown by the line field exactly matches that of the original dimensions.

5.4.2 Conclusions from the Test Simulations

It is interesting to note that the boundaries of the objects can be found using the algorithm explained above. The line field of the image accurately determines the boundary of the image, at least in the simulation. A boundary determination using the line field approach mentioned above can result in easy computation of the periphery, the centroid and the area when the algorithm is applied to the FSR measurements. In the following section, we describe the results obtained using real data and present some discussions on how the results can be improved. We also discuss the pros and cons of the technique and draw relevance between this approach and the heuristic methods described in Chapter 4.

5.5 Results on Real Data

5.5.1 Results

Three objects are tested for this purpose. The objects are described in Table 5.1.

Table 5.1: Test objects used to determine threshold

Object	Weight(gm)	Radius(cm)
Disc I	1860	6.9
Disc II	6860	7.7
RECTANGLE II	1500	14x3.2

The three objects are chosen from two different classes of objects localized in Chapter 4, namely, the circular and the rectangular. The results are only at a preliminary stage and further work needs to be done to conclusively argue the merits and the demerits of the technique. The results with and without the line fields are shown in Fig. 5.10 through to 5.12. In all three cases, the corresponding line field is also marked. The parameters λ_d and λ_g in Eq. 5.21 were changed to accommodate different values of the starting data.

5.5.2 Discussion

Fig. 5.10, Fig. 5.11 Fig. 5.12 & Fig. 5.13 illustrate that the input images have some spikes. The images without using any line field appears to be bigger in size and the entire picture seems to be smooth. The spikes have been averaged out over the neighboring pixels. The implementation of the line field restoration does not significantly change the input image. However, the associated line field image needs some careful considerations. In the following paragraphs, a comparison is made between the two approaches adopted so far, namely, the heuristic ones described in Chapter 4 and the Bayesian restoration described above.

1. The first step of the technique described in Chapter 4 was to threshold the image. This process, as mentioned before, marks the logical boundary between the pixels that are deemed to be in the same class. The thresholding operation, although useful, results in loss of some information by eliminating the pixels that belong to the background from any further processing. Also, the operation

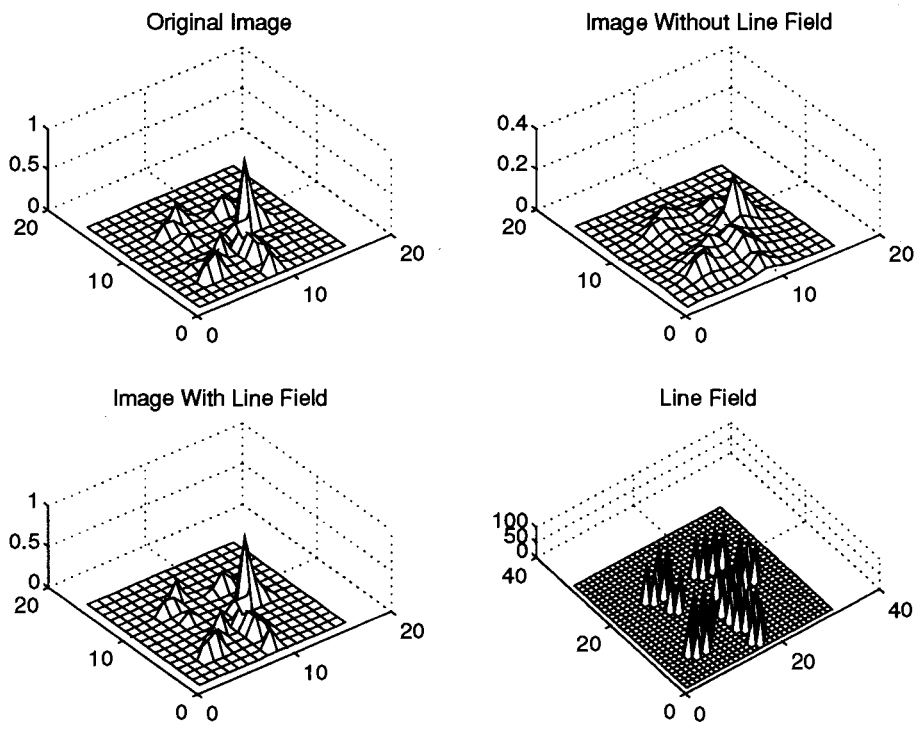


Figure 5.10: DISC I images

results in a binary image, which means that the information about the relative voltages between the pixels are lost. In the Bayesian image restoration approach mentioned above, no such prior filtering step is employed. The image is treated in its grey form and hence this technique can potentially exploit the grey level difference between the pixels in the vicinity of the boundary.

2. The localization techniques mentioned in Chapter 4 utilize prior information about the objects in the form of the shape of its footprint. The object identification scheme described in Chapter 4 suggests an indexing technique that can be used to point to the objects. However, this means that pre-processing is necessary before using the heuristic approaches described in Chapter 4. However, in case of Bayesian image restoration technique described above, no such prior processing is necessary. It can be applied directly to raw images. The results shown in Fig. 5.10, Fig. 5.11 Fig. 5.12 & Fig. 5.13, although not great; still seem to suggest a potential for the Bayesian techniques that can provide equivalent or better results than those suggested in Chapter 4 with some preprocessing of

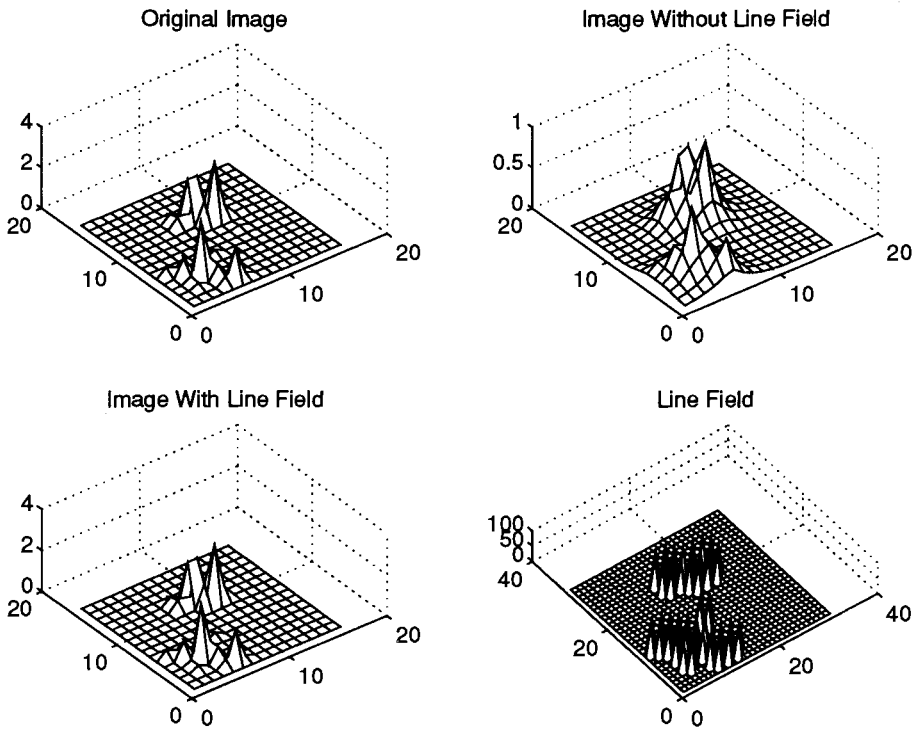


Figure 5.11: DISC II images

post-processing.

3. The processing techniques developed in Chapter 4, assumes that only one object is placed on the array at a time. The techniques will have to be redeveloped for more than one objects. This could be quite a difficult thing if the objects are placed close to each other, owing to the large dimensions of each element and crudeness of the recorded image footprint. The Bayesian techniques developed so far do not make any assumptions about the number of objects placed on the array. The line field plot could potentially differentiate the boundaries between different objects.

4. The Bayesian restoration technique can be improved by incorporating prior knowledge into processing. By assuming prior information about the pressure distribution on the array, some additional information like the heavier side of the object can be obtained.

5. The Bayesian techniques described above are very slow for practical use if applied directly. The program takes anywhere from 15 to 45 minutes to execute,

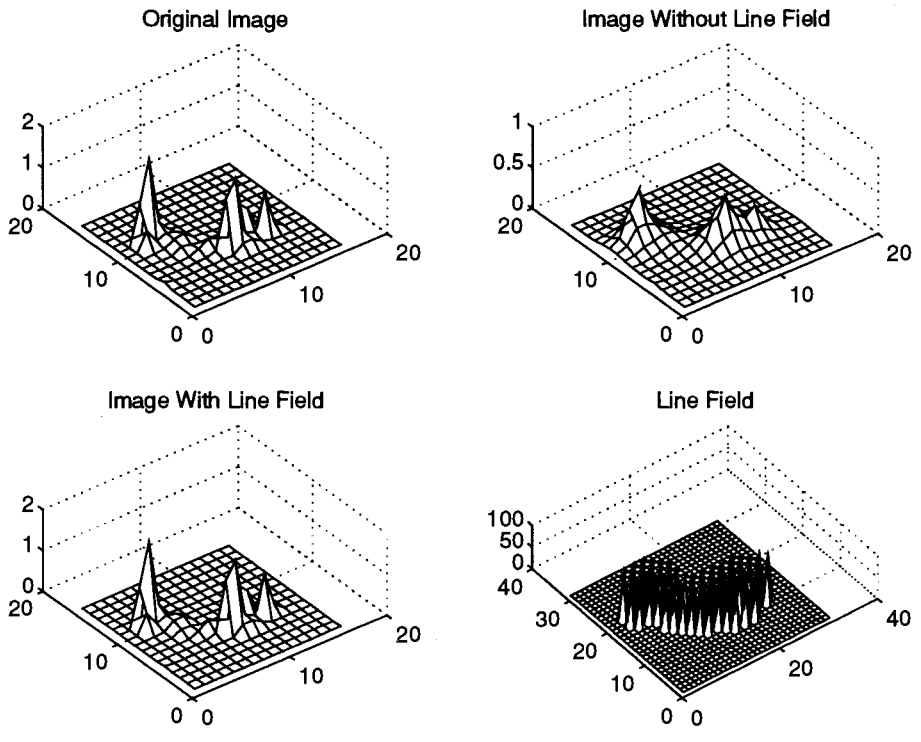
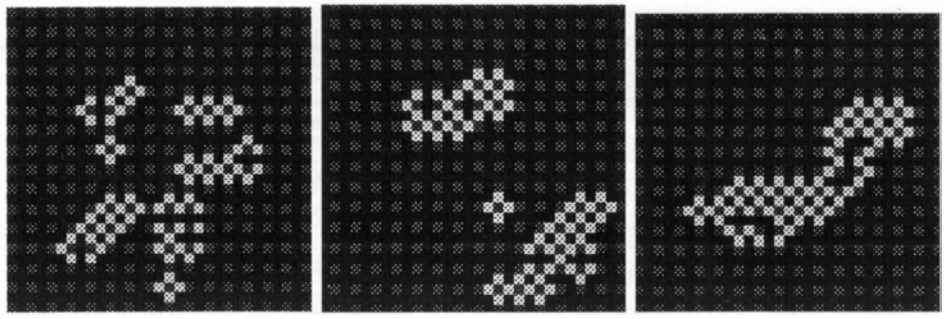


Figure 5.12: Rectangular object RECTANGLE II

however; the processing times can be reduced in direct proportion to the number of processors employed. In other words, the processing times can be reduced by a factor of N be employing N processors at N different sites. Deterministic annealing techniques described in Miller and Rose (1994) can be employed to speed up the process. The technique suggests minimizing energies at each temperature instead of using them to find the stochastic solution process as in case of simulated annealing.

6. As mentioned in 4. above, the results of the algorithm can be improved by incorporating prior knowledge. The potential function defined uses dependencies between the pixels belonging to the neighborhood system. The definition of the potential function could incorporate the prior information into processing. On a similar note, some post-processing can be performed on the results obtained from Bayesian image restoration routines to interpret the results. For example, a convex hull of the boundary found using the line field can force the footprint to follow a convex surface. Table 5.2 shows the results obtained using the convex



(a) (b) (c)

Figure 5.13: Line field images for (a) DISC I (b) DISC II (c) RECTANGLE II

hull on the first two test objects described in Table 5.1

Table 5.2: Result of the convex hull of the line field plots

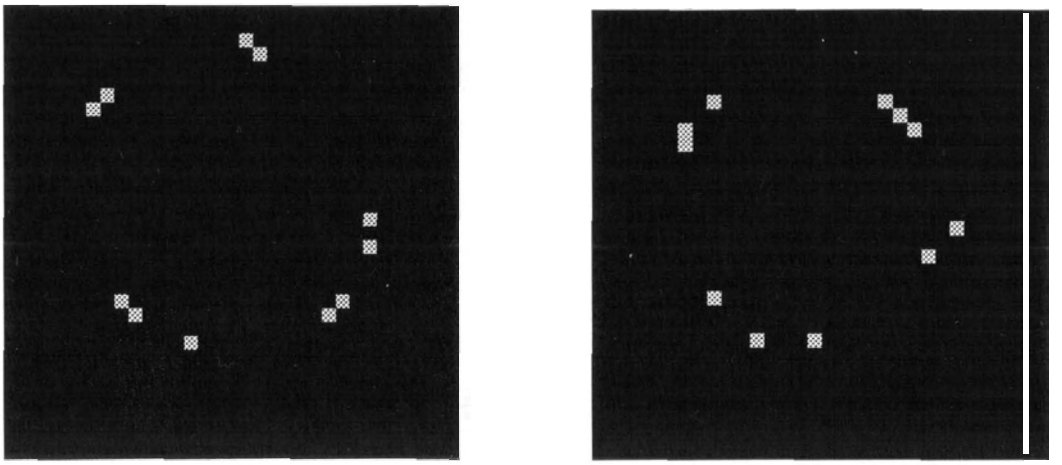
Object	Ra(cm)	Ro(cm)	Roh(cm)
Disc I	6.9	6.3	6.3
Disc II	7.7	7.3	7.3

In Table 5.2 Ra is the actual radius, Ro is the obtained radius after applying convex hull to the line field and Roh is the radius obtained after applying heuristic techniques described in Chapter 4.

It is interesting to see the convex hulls obtained due to the heuristic methods and the convex hull of Fig. 5.13(a) as illustrated in Fig. 5.14.

Fig. 5.14 shows that the convex hulls due to both the methods, namely, the heuristic and the Bayesian techniques are very similar. In fact, as suggested by Table 5.2, the results due to both the algorithms after least squares fit appear to be exactly the same. This can lead us to conclude that the Bayesian technique performs as well as heuristic techniques if post-processing is used.

Summarizing the discussion above, it can be said that the preliminary results using the line field implementation suggests a vast potential for Bayesian image restoration as applied to our case. Incorporating the prior knowledge into processing can provide results which can be expected to be as good as that of the approach mentioned in Chapter 4.



(a) (b)

Figure 5.14: Compare the convex hulls of (a)line field image (b)Due to heuristic techniques in Chapter 4

5.6 Summary

It was seen that the Bayesian techniques provide a set of techniques to filter the data of random noise and also determine the features. Some work needs to be done in future to exploit the full potential of the technique, however.

Chapter 6

Conclusions and Future Work

The purpose of this thesis was to investigate the use of an FSR array as a tactile imaging device. The previous chapters discussed hardware associated with the data acquisition circuitry and described some of the techniques employed to interpret the information from the array. The following sections present some conclusions that can be derived from the array and explain some work that can be carried out based on this thesis.

6.1 Conclusions

6.1.1 Data Acquisition Circuitry

Conclusion The data acquisition circuitry employs a combination of analog and digital circuits. The operation of the circuits is software controlled. The circuits employs discrete components and scans each row and each column using a combination of multiplexers and demultiplexers. The circuit, however, causes electrical coupling between the elements in the same column. This causes errors in the recorded values at each pixel. Also, the total scanning time for the entire array is quite long. It can thus be concluded that the hardware circuit employed is not the optimal solution for the data acquisition and better faster techniques are possible.

Suggested Future Work The possible improvements in the data acquisition circuit can be obtained if a feedback scheme is employed. The voltage generated at the

selected pixel in the array needs to be fed back to the input lines of all the unused rows. This would result in zero potential difference across the elements that are not used and reduces the electrical coupling effects to a minimum. The diagram that depicts the hardware is as shown in Fig. 6.1. The technique has been suggested in (Speeter 1989). The circuit is often called as the site isolation circuit.

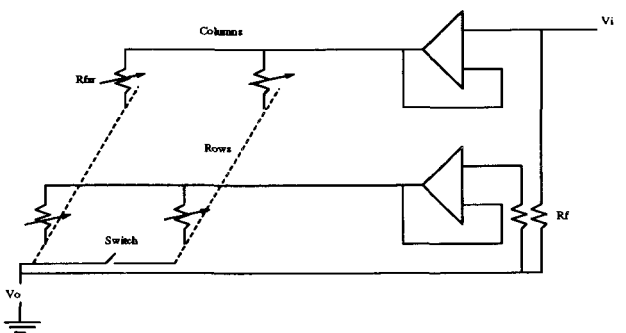


Figure 6.1: Circuit to Counter Coupling Effect

The scanning time can be improved in different ways. Easiest way is to design a dedicated board that performs the operations and interfaces the data with the computer. Faster electronics can be employed. Also, each element is sampled 100 times and the readings are averaged to suppress any noise values that might be recorded. The time can be improved by reducing the total number of samples

6.1.2 The FSR Array

Size and Resolution The size of each element in the FSR array is quite big and hence the readings are quite coarse at this stage. The absence or the presence of an additional element can cause large errors in the object footprint.

An attempt to miniaturize the array can be made. The size of the array is to be made small enough so that it can fit on the fingertip of a robot-hand. This will result in better results and higher resolutions.

Currently, the total number of elements is quite small. Only 16×16 elements are present, which causes a poor resolution in images while using image processing techniques.

The resolution of the array can be improved by using VLSI techniques to shrink the size of the FSR elements. Typically, 16×16 sensors have been fabricated on an area as small as $4\text{in} \times 4\text{in}$ (Christ and Sanderson 1982).

Noise According to our hypothesis, the noise problems with each measurement is due to the presence of dirt, uneven contact surface of the objects and/or the array. These problems cause the readings to be concentrated over a few pixels and cause voids in the actual object footprints.

The noise problem could possibly reduce or at least the effect of the noise could mitigate if the size and the resolution of the array is increased as suggested before. The higher resolution with reduced element size will result in better results as the effect of isolated noise pixels will be reduced and the noise could be filtered more effectively by designing better algorithms.

Pixel Sensitivity The FSR array behaves as a pressure sensor if the object area is larger than the area of each element and is quite sensitive to the position of the object on each element. Also, the readings obtained using multiple objects on top of each other to simulate heavier weights was found to be less reliable compared with a single element. Finally, it was showed in Section 2.4.3 that objects with smaller footprints are recorded more faithfully than those with larger footprints, if the weight per pixel in either case is the same.

It would be interesting to note the effect of heavier objects with smaller areas and compare the results with the curves presented in Section 2.4.3 for the same purpose.

6.1.3 Heuristic Approaches

Conclusions The use of convex hull in locating circular objects is seen to have a vast improvement in the overall result. The heuristic techniques described to locate circular objects work on fairly noisy images and produce results that are still within one pixel accuracy. A bad measurement to begin with results in poor estimate for the radius and the centroids.

It would be a nice exercise to evaluate the performance of these algorithms under varying noise conditions. The recorded measurement can be further corrupted by some known noise and the results can be tabulated. An estimate of the

robustness of the algorithms can thus be determined. The performance of each block can further be evaluated using the same technique. It would be interesting to see the effects in the results by subdividing each pixel into more than 4 pixels. A curve indicating the improvement in the results against subpixeling can be obtained.

The first step of evaluating the orientation for orientation is sensitive to the noise present in the image and produces poor estimate of the orientation for fairly noisy images. The poor orientation result get propagated through to produce a bad elliptic fit for the ellipse.

A direct elliptic fit by fitting ellipses of different sizes and orientations can be tried. The orientation can be made to vary in some fixed steps and for each such orientation, the best elliptic fit can be be found for each orientation. The most probable ellipse would then be the one that generates least error in least squares sense. The techniques will be computationally very expensive but eliminates the sensitivity to noise of the algorithm explained in Chapter 4.

The entire operation can be performed by fitting a rectangle instead of an ellipse. It is of interest to see the improvements obtained over the elliptic fit results.

The object identification sections suggests that the array can be used to identify different objects with a fair amount of accuracy. More extensive tests should be performed to get a wide range of performance results. The current results are quite restricted due to the limitations in the choice of the objects.

6.1.4 Bayesian Image restoration Techniques

Conclusions The Bayesian image restoration technique was implemented and very initial results were presented. The technique seems quite robust to images corrupted with Gaussian noise. The preliminary results obtained suggest that post-processing of the line field image can produce answers similar to heuristic approaches explained in Chapter 4.

More extensive work needs to be done on obtaining the results after pre-filtering and/or post-filtering before using the Bayesian image restoration techniques. There seems a definite scope of improvement upon using prior knowledge into the definition of the potential function. Some experiments need to be performed

to conclusively decide the potential of this approach.

One potential source of improvements could be to model the noise that corrupts our image differently. It was assumed that the measurements were corrupted by Gaussian noise, however, no experiments were conducted to support the claim and no attempt was made to try out other models for noise.

Some other known noise models can be tried out and the results due to each model can be compared. The "best" results can in fact lead to modeling the noise that gets added to the actual measurements. Deterministic annealing can be performed instead of simulated annealing for a faster execution of the technique.

Appendix A

Convex Hull

Some of the convex hull preliminaries are described below. All of the following are considered for points that lie in *d-dimensional* space E^d (Preparata and Shamos 1985, page 90)

Definition 1

Given K distinct points $p_1, p_2, p_3, \ldots, p_k$ in E^d, the set of points

$$p = \alpha_1 p_1 + \alpha_2 p_2 + \cdots + \alpha_x p_k \qquad (\alpha_j \in \Re, \alpha_1 + \alpha_2 + \cdots + \alpha_k = 1) \tag{A.1}$$

represents the affine set generated by the set of points $p_1, p_2, \ldots, p_k$ and p represents the affine combination of the points.

Definition 2

Given a subset L of E^d, the affine hull, aff(L), is the smallest affine set containing L.

Definition 3

Given K distinct points $p_1, p_2, p_3, \ldots, p_k$ in E^d, the set of points that satisfy the following condition in Eq. A.2 is the *convex set* generated by $p_1, p_2, p_3, \ldots, p_k$ and p is the convex combination of $p_1, p_2, p_3, \ldots, p_k$.

$$p = \alpha_1 p_1 + \alpha_2 p_2 + \cdots + \alpha_x p_k \qquad (\alpha_j \in \Re, \alpha_j \geq 0, \alpha_1 + \alpha_2 + \cdots + \alpha_k = 1) \tag{A.2}$$

As seen, a convex set is simply a subset of an affine set where an additional condition of $\alpha_j \geq 0$ needs to be satisfied.

Definition 4

Given an arbitrary convex subset L of points in E^d, the convex hull conv(L) of L, is the smallest convex set containing L.

Definition 5

A *polyhedral set* in E^d is the intersection of a finite set of closed half-spaces (A half space is the portion of E^d lying on one side of a hyperplane.

A bounded *d-dimensional* polyhedral set is referred to as a *a convex d-polytope.*

Theorem 1

The convex hull of a finite set of points in E^d is a convex polytope; conversely, a convex polytope is the convex hull of a finite set of points.

Boundary of a convex polytopes is called as faces and each face is defined on the basis of its dimension. For a *d-polytope*, its $d-1$ faces are called as facets, its $(d-2)$ faces are called as its subfacets, its 1-faces are called as edges and 0-faces are called as its vertices.

Appendix B

Least Squares Fit

Consider a set of N points represented by their Cartesian co-ordinates. To fit a circle in the least squares sense involves minimizing the perpendicular distance of each point from the "best circle" as shown in Fig. B.1.

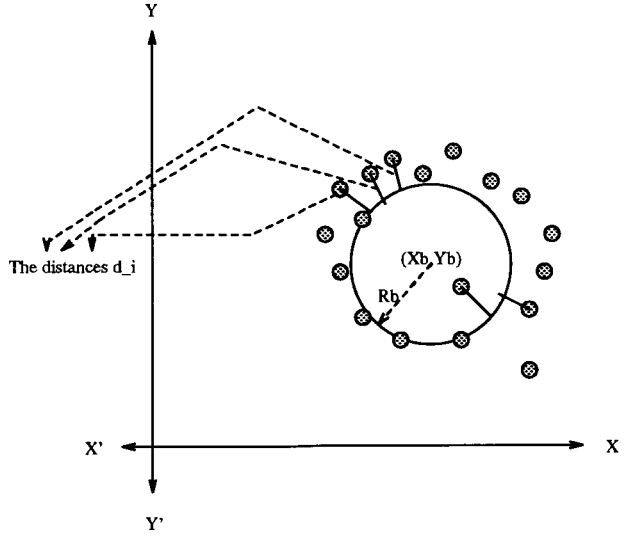


Figure B.1: Visual Description of the distance that is minimized

Let (X_b, Y_b) be the centroids of the "best" circle and let radius R_b be the least squares radius.

Let the set of points be $(X_i, Y_i)\ |_{i=1,N}$.

The distance of each point (X_i, Y_i) from the center (X_b, Y_b) is given by

$$d_i^2 = (X_i - X_b)^2 + (Y_i - Y_b)^2 \tag{B.1}$$

and

$$d^2 = \sum_{i=1}^{N} d_i^2 \tag{B.2}$$

The least squares fit minimizes the sum of the squared errors between the distances d_i in Eq. B.1 and R_b.

Mathematically this can be expressed as

$$e = \sum_{i=1}^{N} (R_b - d_i)^2 \tag{B.3}$$

The error e as expressed by Equation B.3 is a function of the "best" center points (X_b, Y_b) and the least squares radius R_b.

Substituting Equation B.1 into Equation B.3 and simplifying, we get,

$$e = \sum_{i=1}^{N} (R_b)^2 + (X_i - X_b)^2 + (Y_i - Y_b)^2 - 2R_b\sqrt{(X_i - X_b)^2 + (Y_i - Y_b)^2} \tag{B.4}$$

To find the minimum error, we differentiate partially w.r.t. R_b, X_b, Y_b and find $\frac{\partial e}{\partial X_b}, \frac{\partial e}{\partial Y_b}, \frac{\partial e}{\partial R_b}$, shown in Equation B.5, Equation B.6, Equation B.7.

$$\frac{\partial e}{\partial X_b} = \sum_{i=1}^{N} (2(X_b - X_i) - \frac{4R_b(X_b - X_i)}{2\sqrt{(X_i - X_b)^2 + (Y_i - Y_b)^2}}) \tag{B.5}$$

$$\frac{\partial e}{\partial Y_b} = \sum_{i=1}^{N} (2(Y_b - Y_i) - \frac{4R_b(Y_b - Y_i)}{2\sqrt{(X_i - X_b)^2 + (Y_i - Y_b)^2}}) \tag{B.6}$$

$$\frac{\partial e}{\partial X_b} = \sum_{i=1}^{N} (2R_b - 2\sqrt{(X_i - X_b)^2 + (Y_i - Y_b)^2}) \tag{B.7}$$

However, from Equation B.1 $d_i^2 = (X_i - X_b)^2 + (Y_i - Y_b)^2$. Substituting, this in Equation B.5, Equation B.6,& Equation B.7, and simplifying Equations B.8, B.9 & B.10 can be obtained.

$$nX_b - \sum_{i=1}^{N} X_i - X_b R_b \sum_{i=}^{N} \frac{1}{d_i} + R_b \sum_{i=1}^{N} \frac{X_i}{d_i} = 0.0 \tag{B.8}$$

$$nY_b - \sum_{i=1}^{N} Y_i - Y_b R_b \sum_{i=}^{N} \frac{1}{d_i} + R_b \sum_{i=1}^{N} \frac{Y_i}{d_i} = 0.0 \tag{B.9}$$

$$R_b - \sum_{i=1}^{N} d_i = 0.0 \tag{B.10}$$

Starting with some initial guess of $X_b, Y_b \& R_b$ the three equations B.8, B.9 & B.10 can be iteratively minimized by substituting for the new values of R_b, X_b, Y_b till the value in Equations B.8, B.9 & B.10 fall to a low value close to zero.

Appendix C

Convex Hull Algorithm

The algorithm used in the thesis is an FTP copy of the algorithm due to (Bradford Barber, Dobkin, and Huhdanpaa 1993). The technique suggests the combines use of the beneath beyond algorithm and the quickhull techniques used for computing convex hulls in generalized N dimensions. Following sections present a brief explanation of the each of the techniques. More detailed explanation can be found in the book by (Preparata and Shamos 1985).

C.1 QUICKHULL technique

Given a set of points, find the points with maximum and the minimum x coordinates and form a line LR joining the two points. The line divides the space into two regions $S1$ & $S2$ as shown in Fig. C.1. For both the regions, find the triangle with maximum area. By definition, the point P_1, corresponding to the triangle with maximum area, is one of the vertices of the convex hull. Next step is to throw away all the points that lie within the triangle shown by the shaded region in Fig. C.1. Point P_1 gives rise to two more lines, LP_1 & RP_1 and forms two more regions S_{11} & S_{12}.

The above process is repeated for all the points and regions till all the points in the convex hull are covered. The tree for the operation is shown in Fig. C.2.

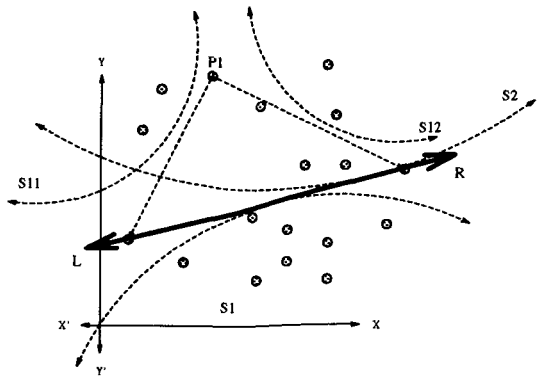


Figure C.1: Quick Hull Technique for Convex Hull

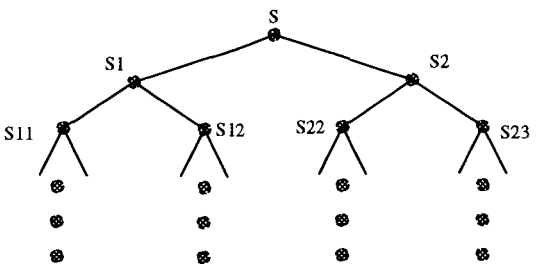


Figure C.2: Quick Hull Technique for Convex Hull

C.2 Beneath-Beyond Algorithm

The algorithm is different from the quick hull technique. Here, a randomized order of the total number of points in the convex hull is created. The technique considers only one point P at a time and examines the conditions on the point. In other words, it checks to see if the point lies within the convex hull or outside of it. If the point lies to the exterior of the convex hull, then, a cone called as the supporting cone is constructed. The cone is seen by joining lines $P4$ & P and P & $P7$. The points lying in the shaded region of the cone are then ignored and the process is repeated for all the points in the set. The technique is called as beneath-beyond because of the way it works. It is based on the following theorems.

Consider a convex hull formed by the union of point P and the existing convex hull PP or let $P' = CH(PP \cup P)$. Then each face of P' satisfies one of the following two conditions.

1. For a face f of PP, there exists a facet F of P such that, $f \subset F$ and P is beneath F. This case is illustrated by face f_1 in Fig. C.3

2. IF f is a face of PP, then $f' = CH(f \cup P)$ is a face of P' *iff* either
   (a) among the facets of P that contain f, there is atleast one such that P is beneath it and at least one such that P is beyond it. This case is illustrated by vertex $f3$ in Fig. C.3
   (b) $p \in aff(f)$ in other words, P can be expressed as a linear combination of f as illustrated by face $f2$ in Fig. C.3.

The detailed operation can be found in the book by (Preparata and Shamos 1985, page 131)

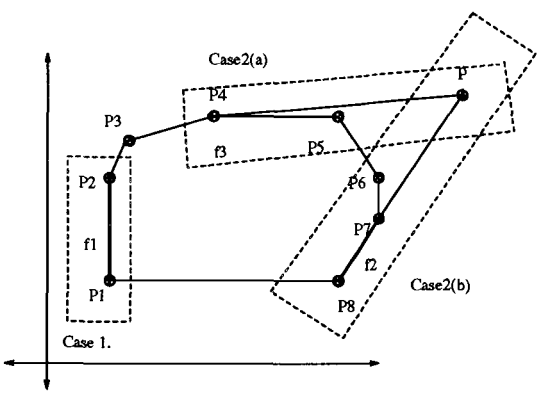


Figure C.3: Quick Hull Technique for Convex Hull

The algorithm used in the thesis makes use of both the above techniques such that the total time, memory usage and the efficiency of the program is improved. Furthermore, it generates convex hull for n-dimensional case.

Appendix D

Least Squares fit for Ellipses

The first step in evaluating the "best" ellipse that fits a given set of data is to form an equation representing the error distances. Distance of any point say p from an ellipse is given by the distance between point p and the point where the normal from point p intersects the ellipse. For simplicity of derivation, a slight modification is made to the above calculation. Since, it is known that the ellipse is always centered at the origin, the distance between point p and the point where the line joining p and the origin intersects the ellipse is used.

The pictorial representation of the distance using the normal to an ellipse and the distance used in our case is shown in Fig. D.1. As can be seen that there is little difference between the two distances as long as the points lie within a close proximity of the "best" ellipse.

The derivation for the error function can be derived as follows.

Consider any point (x_i, y_i). The equation of the straight line passing through(x_i, y_i) and origin is given by

$$y = \frac{y_i}{x_i} x \tag{D.1}$$

Equation of the ellipse centered at the origin can be written as

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \tag{D.2}$$

or

$$x^2 b^2 + y^2 a^2 = 1 \tag{D.3}$$

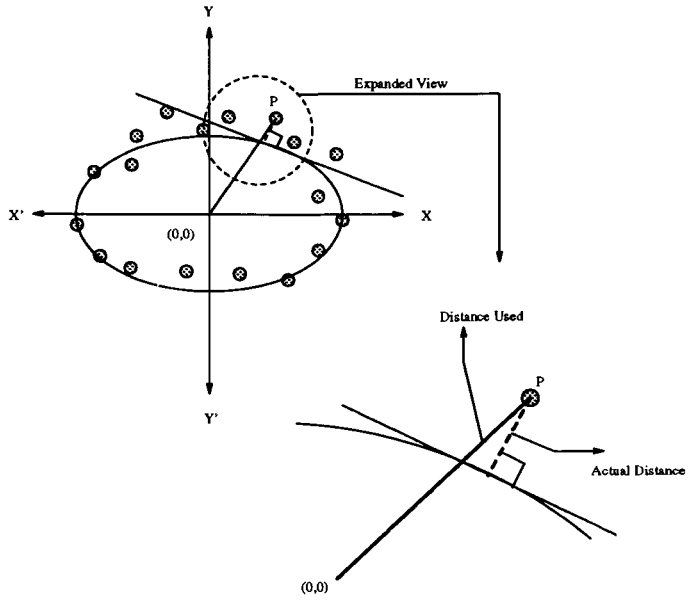


Figure D.1: The error distances used in the calculation

To find out the the points where the line intersects the ellipse, we substitute Eq. D.1 in Eq. D.3 and simplifying for x, we get

$$x^2 = \frac{a^2b^2}{b^2 + \frac{a^2y_i^2}{x_i^2}} \tag{D.4}$$

$$x = \pm\sqrt{\frac{a^2b^2}{b^2 + \frac{a^2y_i^2}{x_i^2}}} \tag{D.5}$$

Substituting Eq. D.5 into Eq. D.1, corresponding y co-ordinates can be found. The values of the y-coordinates is given by

$$y = \pm\frac{y_i}{x_i}\sqrt{\frac{a^2b^2}{b^2 + \frac{a^2y_i^2}{x_i^2}}} \tag{D.6}$$

The pair of points where the line intersects the ellipse is given by

$$\left(\sqrt{\frac{a^2b^2}{b^2 + \frac{a^2y_i^2}{x_i^2}}}, \frac{y_i}{x_i}\sqrt{\frac{a^2b^2}{b^2 + \frac{a^2y_i^2}{x_i^2}}}\right) \quad \&\left(-\sqrt{\frac{a^2b^2}{b^2 + \frac{a^2y_i^2}{x_i^2}}}, -\frac{y_i}{x_i}\sqrt{\frac{a^2b^2}{b^2 + \frac{a^2y_i^2}{x_i^2}}}\right) \tag{D.7}$$

Since, there are two points where the line intersects the ellipse, there are two possible distances between point (x_i, y_i) and the ellipse. However, a look back at diagram

depicted in Fig. D.1 suggests that we are interested in only the shorter of the two distances.

The two distances corresponding to points in Eq. D.7 are given as

$$\begin{aligned} d1_i^2 &= \left(\sqrt{\frac{a^2b^2}{b^2+\frac{a^2y_i^2}{x_i^2}}} - x_i\right)^2 + \left(\frac{y_i}{x_i}\sqrt{\frac{a^2b^2}{b^2+\frac{a^2y_i^2}{x_i^2}}} - y_i\right)^2 \\ d2_i^2 &= \left(-\sqrt{\frac{a^2b^2}{b^2+\frac{a^2y_i^2}{x_i^2}}} - x_i\right)^2 + \left(-\frac{y_i}{x_i}\sqrt{\frac{a^2b^2}{b^2+\frac{a^2y_i^2}{x_i^2}}} - y_i\right)^2 \end{aligned} \tag{D.8}$$

The distances are calculated for all the N points in the boundary such that minimum of the two distances $d1^2 \& d2^2$ are used for calculating the overall distance. Mathematically, the condition is expressed in

$$err = \sum_{i=1}^{N} min(d1_i^2, d2_i^2) \tag{D.9}$$

References

Begej, S. (1984, November). An optical tactile-array sensor. coins 84-26. Technical Report COINS 84-26, University of Massachusetts at Amherst, Dept. of Computer and Information Science.

Berger, A. and P. Khosla (1988). Edge detection for tactile sensing. In *Proceedings of SPIE. INt. Soc. for Optical Engineering*, pp. 163–172. SPIE.

Boie, R. (1984). Capacitive impedance readout tactile image sensor. In *International Conference on Robotics*, pp. 370–378. IEEE: New York.

Bradford Barber, C., D. Dobkin, and H. Huhdanpaa (1993, July). The quickhull algorithm for convex hull. Technical Report Research Report GCG53, University of Minnesota, The Geometry Center, Univ. of Minnesota, USA.

Briot, M. (1979). The utilization of an artificial skin sensor for the identification of solid objects. In *Proc. of the 9th Int. Symposium on Industrial Robotics*, pp. 529–547.

Chalupa, V., V. Marik, and J. Volf (1982). Tactile matrix for shape recognition. In *Proc. of the 9th IMEKO Congress Int. Measurement Confederation*, pp. 339–348.

Checinsky, S. and A. Agrawal (1986). Magnetoelastic tactile sensor. In A. Pugh (Ed.), *Robot Sensors- Vol2 Tactile and Non-Vision*, Chapter 3, pp. 229–235. Bedford, UK: IFS (Publications) Ltd.

Christ, J. and A. Sanderson (1982, September). A prototype tactile sensor. Technical Report CMU-RI-TR-82-16, Carnegie Mellon University, Robotics Institute.

Clark, J. (1990). Split-drain mosfet magnetic sensor arrays. *Sensors and Actuators A 24*, 107–116.

Dario, P., A. Bicchi, and G. Buttazzo (1986). A sensorised scenario for basic investigation on active touch. In A. Pugh (Ed.), *Robot Sensors-Vol2 Tactile and Non-Vision*, Chapter 3, pp. 237–245. Bedford, UK: IFS (Publications) Ltd.

Desai, A., S. Payandeh, and J. Vaisey (1994, October). On the loaclization of objects using an fsr pressure pad transducer. In *IEEE International Conference on System, Men and Cybernetics*, San Antonio, U.S.A., pp. 953–957.

Dougherty, E. and C. Giardina (1987). *Matrix Structured Image Processing.* Eaglewood Cliffs, NJ: Prentice Hall, Inc.

Geman, S. and D. Geman (1984, November). Stochastic relaxation, Gibbs distributions, and the Bayesian estimation of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence 6*(6), 721–741.

Gersho, A. and R. Gray (1992). *Vector Quantization and Signal Compression.* Norwell, MA 02061, USA: Kluwer Academic Publishers.

Ghani, N. (1988a). Esprit 278 - a collaborative sensors project. In *CIM Europe Conference (4th)*, pp. 59–70. Proceedings / CIM.

Ghani, N. (1988b). Visual and tactile senses in collaboration. In *Sensor Review*, pp. 210–215. International Projects. sensor review : 8(4) 1988 ISSN: 0260-2288.

Golub, G. and C. Van Loan (1983). *Matrix Computations.* U.S.A.: The John Hopkins University Press.

Hackwood, S., G. Beni, and T. Nelson (1986). Torque-sensitive tactile array for robotics. In A. Pugh (Ed.), *Robot Sensors-Vol2 Tactile and Non-Vision*, Chapter 2, pp. 123–131. Bedford, UK: IFS (Publications) Ltd.

Hargreaves, D. (1994, August). Interpolation and motion compensation of interlaced video. Master's thesis, Simon Fraser University, Burnaby, Canada, School of Engineering Science.

Harmon, L. (1982, summer). Automated sensing for robots. *The International Journal of Robotics Research 1*(2), 3–32.

Horn, B. (1986). *Robot Vision.* Cambridge, MA: The MIT Press.

Interlink Inc. (1991). *FSR Technical Specifications.* P.O. Box 40760, Santa Barbara, CA 93103: Intelink Electronics Inc.

Jain, A. (1989). *Fundamentals of Digital Image Processing.* Englewood Cliffs, NJ: Prentice Hall.

Jayawant, B. and M. Onori (1986). Robot tactile sensing: A new array sensor. In A. Pugh (Ed.), *Robot Sensors Vol2 Tactile and Non-Vision*, Chapter 2, pp. 199–205. Bedford, UK: IFS (Publications) Ltd.

Konrad, J. (1989, June). *Bayesian Estimation of Motion Fields from Image Sequences.* Ph. D. thesis, McGill University, Montreal, Canada.

Leung, A. and S. Payandeh (1994, October). Application of adaptive neural network to localization of objects using pressure array transducer. In *IEEE International Conference on System, Men and Cybernetics*, San Antonio, U.S.A., pp. 2114–2119.

Liu, H. and P. Meusel (1993, June). Development of a new kind of robot slipping sensor. In *Proc. of the American Control Conference*, pp. 756–757. ACC.

Miller, D. and K. Rose (1994, February). Combines source-channel vector quantization using deterministic annealing. *IEEE Transactions on Communications 42*(2), 347–359.

Nicholls, H. and M. Lee (1989, June). A survey of robot tactile sensing technology. *The International Journal of Robotics Research 8*(3), 3–30.

Nowlin, W. (1991, April). Experimental results on bayesian algorithms for interpreting compliant tactile sensing data. In *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, pp. 378–383.

Pratt, W. K. (1991). *Digital Image processing* (2 ed.). John Wiley and Sons, Inc.

Preparata, F. and M. Shamos (1985, May). *Computational Geometry: an Introduction.* New York: Springer Verlag.

Rebman, J. and K. Morris (1986). A tactile sensor with electrooptical transduction. In A. Pugh (Ed.), *Robot Sensors-Vol2 Tactile and Non-Vision*, Chapter 2, pp. 145–155. Bedford, UK: IFS (Publications) Ltd.

Robertson, B. and A. Walkden (1986). Tactile sensor system for robotics. In A. Pugh (Ed.), *Robot Sensors-Vol2 Tactile and Non-Vision*, Chapter 2, pp. 89–97. Bedford, UK: IFS (Publications) Ltd.

Russell, R. A. (1990). *Robot Tactile Sensing.* Australia: Prentice Hall.

Sato, N. and A. Pugh (1986). A method for three-dimensional part identification by tactile transducer. In A. Pugh (Ed.), *Robot Sensors-Vol2 Tactile and Non-Vision*, Chapter 2, pp. 133–143. Bedford, UK: IFS (Publications) Ltd.

Speeter, T. (1989). Flexible, piezoresistive touch sensing array. *Proc. of the SPIE. The Int. Soc. of optical Engineering 1005*(Optics, Illuminations and Image Sensing for Machine Vision III), 31–43.

Van Brussel, H. and H. Belien (1986). A high resolution tactile sensor for part recognition. In *Proc. of the 6th Int. Conf. On Robot Vision And Sensory Controls*, Kempston, UK, pp. 49–60. IEEE.

Yaniger, S. (1991). Force sensing resistors: A review of the technology. In *Electro International Conference record*, pp. 666–668. Interlink Electronics, Inc.