AN AUTOMATIC VISION GUIDED POSITION CONTROLLER IN A CONVEYOR BELT PICK AND PLACE SYSTEM

A Thesis Submitted to
the Graduate School of Engineering and Sciences of
İzmir Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE

in Computer Engineering

by
Özgür ÇELİKDEMİR

October 2006
İZMİR

We approve the thesis of **Özgür ÇELİKDEMİR**

Date of Signature

.... 10 October 2006

Prof. Sıtkı AYTAÇ
Advisor
Department of Computer Engineering
İzmir Institute of Technology

.... 10 October 2006
Asst.Prof. Şevket GÜMÜŞTEKİN
Co-Advisor
Department of Electrical and Electronics Engineering
İzmir Institute of Technology

.... 10 October 2006
Prof. Erol UYAR
Department of Mechanical Engineering
Dokuz Eylül University

.... 10 October 2006
Prof. Halis PÜSKÜLCÜ
Department of Computer Engineering
İzmir Institute of Technology

.... 10 October 2006
Asst. Prof. Zekeriya TÜFEKÇİ
Department of Computer Engineering
İzmir Institute of Technology

.... 10 October 2006
Prof. Kayhan Erciyeş
Head of Deparment
Department of Computer Engineering
İzmir Institute of Technology

....
Assoc.Prof. Semahat ÖZDEMİR
Head of the Graduate School

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to my advisor Prof. Sıtkı Aytaç for his support. I would like to express my sincere thanks to my co-advisor Asst.Prof. Şevket Gümüştekin for his invaluable support, guidance and patience in the preparation of this thesis.

I would also like to thank my committee members Prof. Erol Uyar, Asst.Prof. Zekeriya Tüfekçi and Inst. Tolga Ayav for their support and advice.

I am grateful for my work mates who compensated for my absence at work. I would like to also thank to Project Manager Hüseyin Özkeser for his support and understanding.

I am very grateful for my parents for their invaluable continuous support.

ABSTRACT

AN AUTOMATIC VISION GUIDED POSITION CONTROLLER IN A CONVEYOR BELT PICK AND PLACE SYSTEM

An automatic vision guided position controller system is developed as for possible applications such as handling and packaging that require position and orientation control. The aim here is to minimize the production cycle time, and to improve the economic performance and system productivity. The system designed can be partitioned into five major parts: vision module, pneumatic automation module, manipulator, conveyor-belt and a software that manages and integrates these modules.

The developed software captures raw image data from a camera that is connected to a PC via usb port. Using image processing methods, this software determines the proper coordinates and pose of the moving parts on the conveyor belt in real time. The pick and place system locates the parts to the packaging area as part's predefined orientation. The software communicates with a controller card via serial port, manages and synchronizes the peripherals (conveyor belt- stepper motors- pneumatic valves,etc) of the system. C programming language is used in the implementation. OpenCV library is utilized for image acquisition.

The system has the following characteristics: The Conveyor belt runs with a constant speed and objects on the conveyor belt may have arbitrary position and orientation. The vision system detects parts with their position and orientation on the moving conveyor belt based on a reference position. The manipulator picks the part and then corrects its position comparing the information obtained by vision system with predefined position, and it places the object to the packaging area. System can be trained for the desired position of the object .

ÖZET

KONVEYÖR BAND TUT/YERLEŞTİR SİSTEMİNDE OTOMATİK GÖRSEL YÖNLENDİRMELİ BİR KONTROL MEKANİZMASI

Bu çalışmanın amacı, posizyon ve konum kontrollu gerektiren "tut ve yerleştir" uygulamalarında, üretim operasyonlarına yardımcı, üretim sürecini kısaltan ve verimliliğini arttıran, üretime ekonomik katkı sağlayan bir prototip geliştirmektir. Tasarlanan sistem genel olarak yapay görme modülü, pnömatik otomasyon modülü, manipulator, konveyor bant, tüm bu modüllerin kontrolünü ve kordinasyonunu sağlanayan bir yazılımdan oluşmaktadır.

Geliştirilen yazılım, bilgisayara usb portu ile bağlı bulunan bir web kamera sayesinde konveyör üzerinde hareket eden parçanın işlenmemiş-ham görüntüsünü alır ve görüntüsü alınmış olan parçanın koordinat ve pozisyon bilgisini görüntü işleme teknikleri kullanarak gerçek zamanlı olarak analiz ve tespit eder. Daha sonra "tut yerleştir" sistemi, analiz edilen parçayı daha önceden belirlenmiş konumuna göre düzelterek, paketleme bölgesine koyar. Konum pozisyonlaması step motor ile sağlanmıstır. Yazılım seriporta bağlanan harici bir kontrol kartı ile haberleşerek sistem çevresel modüllerini yönetir ve çalışmalarını senkronize eder. Yazılım dili olarak C programlama dili kullanılmıştır.

Sistemin sahip oldugu karakteristikler şu şekilde sıralanabilir: Konveyör-bant sistemi sabit hızda hareket etmektedir. Konveyör üzerindeki parçalar rastgele bir konumda ve pozisyonda olabilir. Yapay görme sistemi parçaların konum ve pozisyon bilgilerini daha önceden belirlenen bir refereans noktasına göre tespit eder. Manipulatör parçayı yakalar, daha önce tanımlanmış pozisyon bilgisi ile karşılaştırarak posisyonunu düzeltir ve paketleme bölgesine istenilen pozisyonda koyar. Parçanın paketleme bölgesine koyulması gereken posizyon, sisteme tanımlanabilir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Robot manipulators are widely used in today's industrial world. There are various manipulator systems for different tasks and different industrial applications "(Geisselmann et al. 1986)". They are commonly used for highly specialized work "(Mitchell 1987)" functions because, they can perform repetitive tasks more quickly, inexpensively, and accurately than humans. Moreover they never get tired and their performance is predictable. These features make robots preferable for industrial applications.

Robot manipulators become more useful and effective by carefully chosing the right sensors "(Plenum Press 1979)". Machine vision is an important form of perception for robotic applications. Part identification, defect inspection, presence-absence detection, dimension measurement and positioning are some application areas of machine vision in industry "(Rosen 1979)". The extracted information from these applications can be used for the purpose of controlling a robot manipulator for handling, assembly, welding, etc "(Hollingum, 1984)". A vision-guided production system that installs a cover on an arbitrarily positioned cell phone can be given as an example. Some of the other existing applications of robotic guidance are, automotive windscreen alignment and placement "(Hollingum 1984)"; high precision part assembly in aerospace applications "(Simith and D. Johnson 1989)"; pattern-correct sewing in textile manufacture "(Geisselmann et al. 1986)"; vision guided nuclear fuel sub-assembly dismantling "(Todd 1988)".

The purpose of this work is to develop a vision guided position and orientation controller for a pick and place system, for use in handling and packaging facilities and applications that require position and orientation control. The system is designed to determine position and orientation of parts which are randomly placed on a moving conveyor belt and also to place them to the packaging area with the desired orientation. System has the following characteristics: Conveyor belt system runs with a constant speed. The objects may have different position and orientation on the conveyor belt . The vision system detects parts with their position and orientation on the moving conveyor belt with respect to a reference position. This system performs mutiple tasks.

First, the manipulator picks the part and then corrects the position of this part to match the predefined position and finally places it to the packaging area. System can be trained for different positions. The system designed can be partitioned into five major parts: vision module, pneumatic automation module, manipulator, conveyor-belt and a software that manages and integrates these modules. Detailed information about these modules are discussed in chapter 2.

The image processing part of the software is given top priority. The image data is captured by using openCV® “(WEB_1 2006)”. An image is subtracted from background and binarisation is achieved by applying gaussian filter, histogram analysis, threholdings “(Galbiati 1990)”. After these preprocessing operations, image is segmented into regions with different pixel values, by applying a connected component labelling algorithm. These image processing methods are detailed in chapter 3. To investigate the position of the part, the mass centers of these regions are used. For objects that has only one region, boundary based methods are used to find the position. These position finding methods are explained in chapter 4.

Also region based and boundary based shape descriptors “(Galbiati 1990)” are disscussed in chapter 3. The major focus of the design is on finding the correct position and orientation finding and synchronizing the manipulator’s modules and their movements.

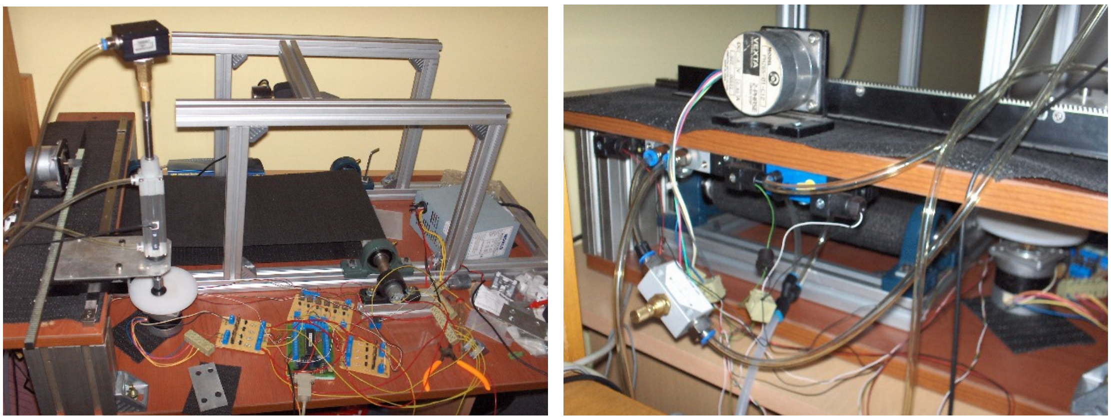
Design objectives of this work can be summarized as follows;

- Developing a prototype for use in handling and packaging facilities and applications that require position and orientation control.
- Implementing a software to control and synchronize the movement of the robot manipulater based on images acquired.
- Combining the computer software, mechanical engineering and electronics as an interdisciplinary study.
- Gaining experience in engineering design process including planning, manufacturing, ordering materials, assembling the parts, hardware and software implementation and testing.

CHAPTER 2

DESCRIPTION OF MECHANICS AND COMPONENTS USED

The system consists of many hardware components which can be partitioned into five modules: Vision, pneumatic, mechanical, electronic and software. In the vision module, an inexpensive usb camera is used to retrieve the images. In the pneumatic automation module, a double acting hollowed-rod cyclinder, a suction cup, four valves, a vacuum ejector and an air reservoir are used for handling operation. In the mechanical module, rack and pinion, rail and block are used for translationing the handling system and rotationing the object. Four bearings are used to bear rollers. In the electronic module, two step motors and an electronic controller card (micronet) are used to control motors and valves. Additional electronic circuits are also used to protect the controller card. Two power supply unit is used in the system. A computer (1500Mhz Celeron CPU) is used as the main controller. A simple 12V DC motor is used to drive the conveyor-belt.



(a) (b)

Figure 2.1. General view of the system. (a) Front view of the system. (b) Reverse view of the system.

2.1. Detailed Features of The Components

2.1.1. Vision Module

A standart usb camera is prefered because it does not require much power and also can be purchased relatively cheap. It is connected directly to the computer’s usb port. No additional graber card is needed to retrive the image data. The camera is capable of capturing up to 30 frames per second with a resolution of 640 x 480 pixels.

2.1.2. Pneumatic Module

2.1.2.1. Pneumatic Components

Pneumatic cyclinders are being used in different areas of the industy for different applications and their usage areas are constantly growing. Packaging, transporting, pushing and many other processes can be done easily with the help of these equipments.

Cylinders have two basic functions; Single-acting and double acting ”(Mannesman Rexroth, 1998)”. A single acting cylinder is driven by air pressure in one direction and by a spring in other direction. Single acting cyclinders can perform work in the direction that is given by the air. A double acting cyclinder is driven by air pressure in both directions.They can work in both directions.

(a) (b)

Figure 2.2. Basic cyclinder types and symbols (a) Single Acting Cyclinder (b) Symbol Double Acting Cyclinder

Basicly, the dimensions of a cyclinder is determined according to the required force. The force of the cyclinder “(Mannesman Rexroth 1998)” is :

$$F = P \times A \tag{2.1}$$

Where,

F, Force in kg

P, Pressure in bar

A, AnnulusArea, cm^2

In this system double acting- hallowed rod cylinder is used. The cyclinder's piston diameter is Ø 25mm, stoke is 50mm, maximum permissible pressure is 10 bar.

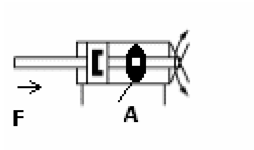
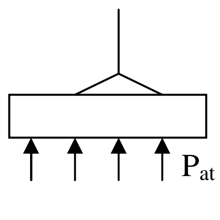


Figure 2.3. Double acting - hollowed rod cylinder

Using formula (2.1), the lifting force is calculated as 24.7 kg at 6 bar theorical operating pressure. Although this lifting force is much greater than what we required (we require max. 0.5 kg lifting force), these cylinders are preferrable since the alternative mini-cyclinders are more expensive and not commonly available.

The reason of using hollowed-rod cylinder is to be able to perform negative pressure(vacuum) at the suction cup.

The suction cups are prefered in pick and place systems. The gripper in vacuum systems is the suction cup. The atmospheric pressure effects the object surface. The suction cup performs the holding force by the difference in pressure between atmospheric pressure and the pressure in the suction cup. In this system, a 25mm diameter suction cup is used.



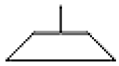
(a)

(b)

Figure 2.4. Suction Cup illumination (a) Lifting illumination (b) Suction cups

Table 2.1. Techical values of suction cups by diameter

(Source: Mannesman Rexroth 1998)

Symbol	External diameter D [mm]	Holding force [N]* (lbf)	Volume [cm^3]	Min. curving on object R_{min} [mm]	Weight [kg] (lbs)
	15	9,0 (2.02)	0,4	9,0	0,0045 (0.010)
	20	15,5 (3.48)	0,8	13,0	0,0056 (0,012)
	25	**26,5 (5.96)**	**1,3**	**17,5**	**0,0085 (0,019)**
	30	34,0 (7.64)	1,3	26,0	0,0092 (0,020)
	35	44,0 (9.89)	2,7	31,0	0,0119 (0,026)
	40	57,7 (13)	3,8	37,0	0,0135 (0,030)
	50	91,0 (20.5)	7,0	41,0	0,0173 (0.038)

*Theoretical value at 60% vacuum level, without safety factor.

Using holding force formula;

$$F = \frac{m \times g}{n} \times s \qquad (2.2)$$

Where,

F, holding force , N

m,weight of object, kg

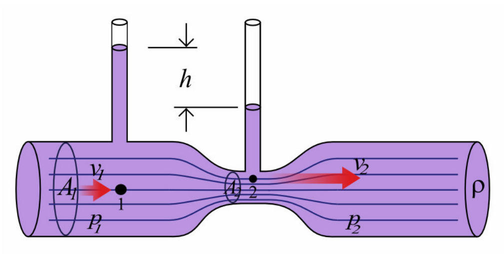
g, gravity factor, 9.81

n, number of suction cups,

s, is safety factor,

In the Table 2.1. it can be seen that the holding force is 26.5 N at %60 vacuum level for choosen suction cup. With the safety factor 5 , permissible mass weight to be lifted is 0.540 kg. This is sufficent for our test object's weight.

The ejector produces negative pressure in vacuum techology. It works like a vacuum pump. A high vacuum is created by the venturi princible. When the ejector evacuates the suction cup, needed holding force is created to handle the objects. The inlet pressure is very important to create a vacuum. Optimum inlet pressure(P_{opt}) is in 4-6 bar range. When the inlet pressure(P_{in}) is lower ($P_{in} < P_{opt}$), the ejector can not create high level of vacuum "(Bosch Rexroth 2003)". In this system EBP-60 is used.



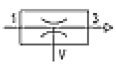
(a)

(b)

Figure 2.5. Vacuum illumination (a) Venturi princible (b) Vacuum ejector

Table 2.2.Techical values of vacuum ejectors by types.

(Source: Mannesman Rexroth 1998)

Symbol	Type	Suction capacity against atmosphere [l/min] (SCFM)	Air consumption at P.opt [l/min] (SCFM)	Sound level without silencer [dBA]*	Sound level with silencer [dBA]*	Weight [kg] (lbs)
	EBP-15	8 (0.28)	15 (0.53)	62	53	0,060 (0.132)
	EBP-30	18 (0.64)	30 (1.06)	75	58	0,080 (0.176)
	EBP-60	**36 (1.27)**	**60 (2.12)**	**66**	**58**	**0,130 (0.287)**

Pneumatic valves are needed to control the air flow "(Mannesman Rexroth 1998)". In this system, 2 directional valves and 2 flow valves are used.

The directional valves are used to control the start, stop and direction of a flow. For controlling the cyclinder up and down movement a 5/3, 24V electrical operated valve with two selenoids is used. 5/3-valve means the valve has 5 ports and 3 positions.

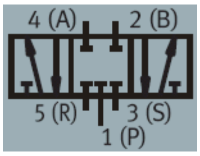


Figure2.6. 5/3-Valve

Port1 is the supply port for air pressure.

Port2 and Port4 are the user ports that make the cyclinder move up-down.

Port3 and Port5 are the exhausts ports.

For controlling the vacuum ejector a , 4/2, 24V electrical operated valve with one selenoid is used. 4/2-valve means the valve has 4 ports and 2 positions.

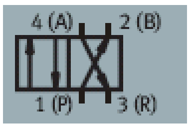


Figure 2.7. 4/2-Valve

Port1 is the supply port for air pressure.

Port2 and Port4 are the user ports. Port2 is closed to block air flow. Port4 is the outlet to the ejector. Port3 is the exhausts ports.

For controlling the cylinder speed, adjustable one-way flow control valves are used. One-way valve permits free flow in one direction. In the opposite direction the valve throttles the flow.

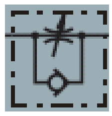


Figure 2.8. One-way flow control valve, adjustable

The general method to control cylinder speed is to restrict the air that exhausted from the cyclinder. Generally it is undesirable to restrict the air going in to cyclinder. In this system, the air leaving from the cyclinder is restricted. This method increases the system’s spring constant and obtains a good control performance.

To supply compressed air, a compressor with 50Lt Reservoir is used for the pneumatic system.

2.1.2.2. Pneumatic Diagram

The pneumatic block diagram is shown in figure 2.8.

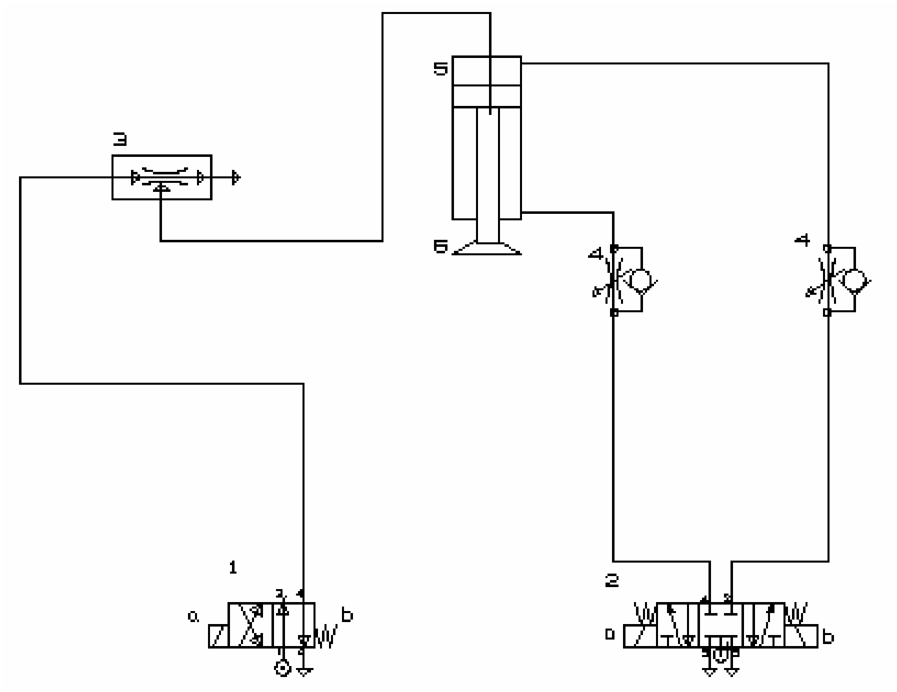


Figure 2.9. Pneumatic diagram of the system

Valve “1” is an 4/2, 24V electrical operated valve with one selenoid. This valve controls the vacuum ejector. In de-energizing state, port 1 is opened to port 3. Port3 is always closed. Port 4 is opened to port 2 that exhaused to atmosphere. No vacuum and holding force occurs in that state. When the coil “a” is energiezed, port 1 is opened to port 4 and the compressed air goes in to the vacuum ejector(3) and creates holding force by venturi principle at suction-cup(5).

Valve “2” is a 5/2, 24V electrical operated valve with two selenoid. This valve controls the cyclinder up and down movement. When both of the coils are de-energized the air is isolated from the cyclinder. If the coil “a” is energized, port1 is opened to port 4 and compressed air goes into the cyclinder to move it up. In other case, when the coil “b” is energized, port 1 is opened to port 2 and compressed air goes into the cyclinder’s opposite side to move it down. While the compressed air is going to the cyclinder’s one side, the air in the other side is exhaused to the air. Otherwise, the cyclinder will be blocked.

2.1.3. Mechanical Module

A rack and pinion system is used to convert rotational movement to lineer movement. Pinion has 12 toothed with 15mm diameter and rack has 500mm length. Pinion is assembled to stepper motor’s shaft.

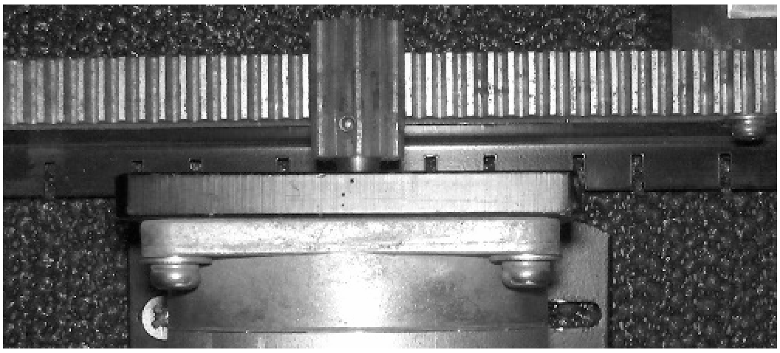


Figure 2.10. Rack and pinion

Rail and block system is used to obtain lineer bearing.

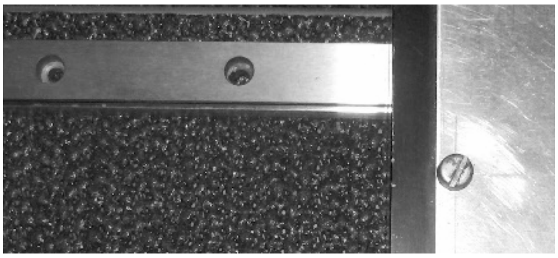


Figure 2.11. Rail and Block

In industrial applications, conveyor belts are usually driven with 3 phases-230V AC electrical motors. A standart electrical motor revolution is 1450rpm. If it is assemblied directly to the rollers, conveyor rollers speed will be 1450rpm. This is very high speed for rollers and bearings. The rotation speed is reduced by using reductors. Reductors also provide increased torque values for the system.

The relation between power, tork and revolution is "(Bosch Rexroth 2003)" :

$$P = \frac{T \times n}{9549} \quad (2.3)$$

P, power ,kw

T, ...,Nm

N,......rpm.

According to formula (2.3) , if an 0.75 kw electric motor is used at revolition n=1450 rpm, the tork value(T) is calculated, 4,94Nm. This rpm is reduced by using reductor ratio of 1:10 to n=145rmp, and the tork value will be, T=49.4Nm.

In this work, neither high speed nor high tork value is necessary. Therefore, a simple DC electric motor is used without using a reductor. The motor's revolution speed is 15 rpm. and at that speed conveyor belt speed is 35 mm/sec.

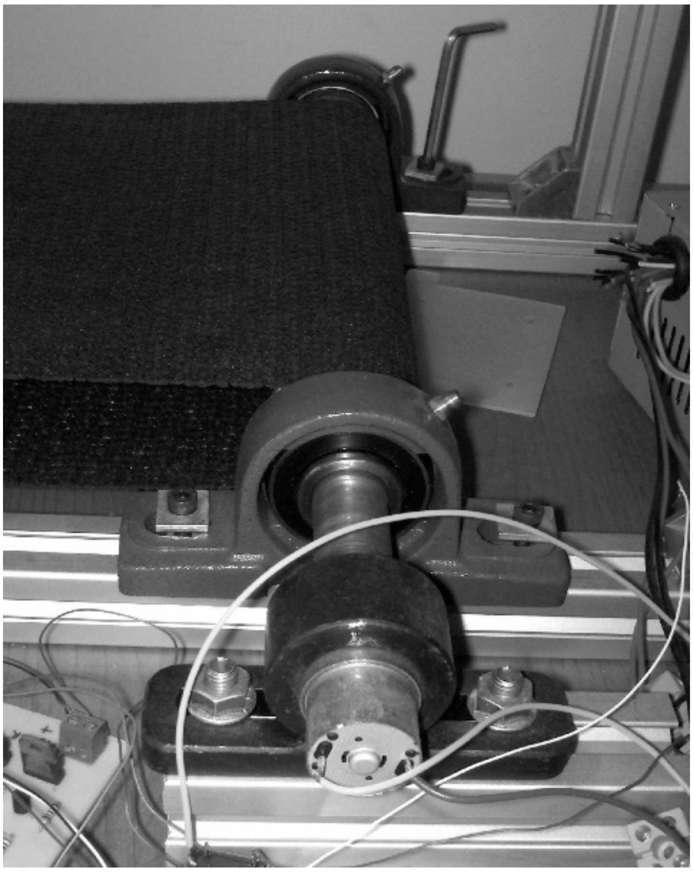


Figure 2.12. Roller bearing and conveyor

2.1.4. Electronic Module

2.1.4.1. Stepper Motors

Stepper motors are used in various applications, such as machine tools, computer peripherals, robotics. Stepper motors convert electronic pulses into mechanical movement. Each electronic pulse causes the shaft to rotate a certain number of degrees.

Stepper motors “(Acarnley, 2002)” have many features that make them popular.

Brushless - Steppers are brushless. Brushes create sparks, undesirable in expolosive environments.

Holding Torque - Steppers have very good low speed and holding torque. Steppers can even hold a position without power applied, using magentic 'detent' torque.

Open loop positioning - Perhaps the most important and interesting feature of a stepper is the ability to position it self without need any feed back, which means they can run 'open-loop' without the need for any kind of encoder to determine the shaft position. They go a certain distance at a certain speed without the need for any type of feedback signal. Closed loop systems are known as *servo* systems. Compared to servos, steppers are very easy to control, the position of the shaft is guaranteed as long as the torque of the motor is sufficient for the load, under all its operating conditions.

Load Independent - The rotation speed of a stepper is independent of load, provided it has sufficient torque to overcome slipping. The higher rpm a stepper motor is driven, the more torque it needs.Slipping is usually a disaster for steppers, because the position of the shaft becomes unknown. For this reason, software usually keeps the stepping rate.

There are two basic types of steppers; Bipolar and unipolar. In this work unipolar stepper motors are used.

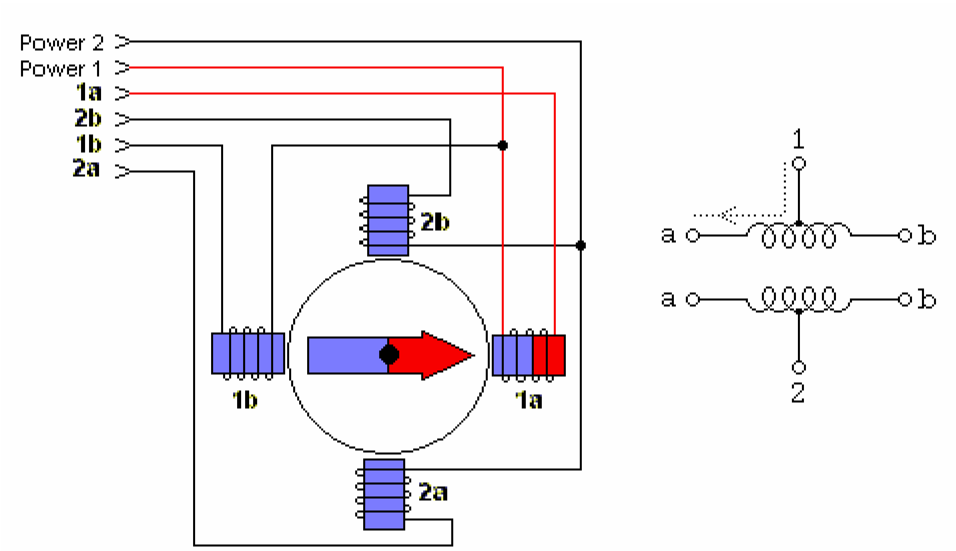


Figure.2.13. Step motor coils illumination. 90 Deg/Step.

The sequence pattern is represented with 4 bits, a '1' indicates an energized winding. After the last step in each sequence the sequence repeats. Stepping backwards through the sequence reverses the direction of the motor.

Full Step Mode :

The stepper motor uses a four-step switching sequence. In full step mode, stepper generates respectively Hi Torque.

Switching Squenences

1a	2b	1b	2a
0	0	1	1
0	1	1	0
1	1	0	0
1	0	0	1

Half Step Mode :

The stepper motor uses a eight-step switching sequence. In half step mode, stepper doubles the stepping resolution of the motor, but the torque is not uniform for each step.

Switching Squenences

1a	2b	1b	2a
0	0	0	1
0	0	1	1
0	0	1	0
0	1	1	0
0	1	0	0
1	1	0	0
1	0	0	0
1	0	0	1

In this work stepper motors are preffered. Because, they have certain positioning without need feedback signal. Also it is easy to drive a stepper motor with simple control system. They are relatively inexpensive.

Two stepper motors are used. One is used to drive rack and pinion for translation of the handling system(Translationing stepper). Other is used to rotate the object in desired angle(Rotationing stepper). Both of the steppers are 6V- 1.8Deg/Step and driven with full-step mode to gain high/consistent torque.

Wire Connections :

Translation Stepper				Rotation Stepper			
1a	2b	1b	2a	1a	2b	1b	2a
Red	Green	Blue	Black	Yellow	Orange	Red	Blue

Detailed connection diagram is illustrated in figure 2.16 .

2.1.4.2. Microcontroller Card

The controller card (Micronet) used here has 8 analog channels and 22 digital channels with 10 bit resolution. Every digital channel can be set as either input or output.

Micronet performs predefined commands via communicating with serial port (RS232) and sends back a response string that reports whether the command is performed or not. It can receive input voltage values in 8 analog channels separately and observes logical status of the buttons (sensors connected to the digital channels that were set as input and/or output). This controller card can be used to read data or to control other peripheral devices.

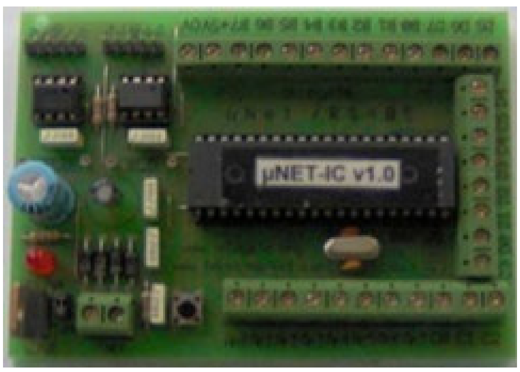


Figure 2.14. Micronet controller card.

Micronet Controller Technical Specifications:

Chip : PIC16F87x

Analog Channel Voltage Supply: 0-5V

Analog Channel Input Empedance: Maximum 10K

Analog Channel Unit/Resolution : 8 units, 10bit resolution.

Digital Channel Specifications: There are totaly 22 digital channels that can be set as either input or output in 3 digital ports. Inputs support TTL, outputs support CMOS level. Every channel feeds 20mA. 3 digital ports consist of B,C,D.

Main Voltage : 5VDC.

Command Process Speed: 52 Command./sec at 9600 bps.

Commands:

Command string format of the controller is "C1 C2 C3 C4 C5 C6 C7 C8".

C1 is always "<". Gets ready to wait command string ;

C8 is always ">". Command is operated after ">" character is entered. ;

C2C3 identity;

C4 is command character. ,

"A" : Reads analog data.

"R" : Reads 1 byte data from digital port.

"W" : Writes 1byte data to digital port.

"P" : Determines input and outputs at digital port. If bit value is "0" port is set as output. If bit value is "1" port is set as input.

"I" : Reads 1 bit data from digital port.

"O" : Writes 1 bit data to digital port.

C5 is analog or digital channel number;

C6C7 is Hex value of the 1 byte data to be written.

For example ;

If the 2.channel of the digital port "D" is to be set as logic "1", the command must be, <33OD21>.

In the same way, if the 2.,4.,7. channels of the port "D" is to be set as logic "1" ,the command mut be,

<33WD94>

94 is the HEX value binary **1**00**1**0**1**00 (**d7** d6 d5 **d4** d3 **d2** d1 d0) .

33 is the identity of the controller.

Port Configration:

3 Digital ports are configrated as listed below;

All the channels of port B except b7 is set as output. Channels b0,b1,b2,b3 are programmed to control rotationing stepper motor. Channel b7 is set as input to receive signal from magnetic cyclinder sensor.

All the channels of port "C" is set as output. Channels "c0,c1,c2,c3" are programmed to control translationing stepper motor. All the channels of port "D" is set as output. Channel "d1" is responsible for controlling conveyor motor. "d2" controls directional pneumatic valf of the vacuum ejector. "d3" contols cyclinder-up movement. "d5" controls cyclinder-down movement.

Port connection diagrams are shown in figure 2.16.

2.1.4.3. Switching Circuits

There are three switching cards used to protect microcontroller card and to be able to use different power source values. Switching cards that control stepper motors use 5V power source. The other card that controls pneumatic valves, magnetic sensor and conveyor drive motor use 24V power source. Figure 2.15. shows the electronic circuit schema for one of switching line.

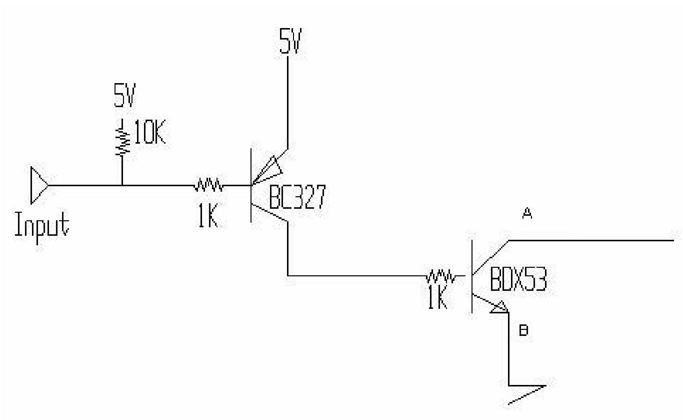


Figure 2.15 Switching circuit diagram

When input signal is +5V, line A is connected to line B which is ground.

NPN and PNP transistors are used for switching. The transistor basically controls a secondary circuit with a small current or voltage in another primary circuit "(Acarnley 2002)".

Detailed connection diagram of the cards,microcontroller and peripherals are in figure 2.16.

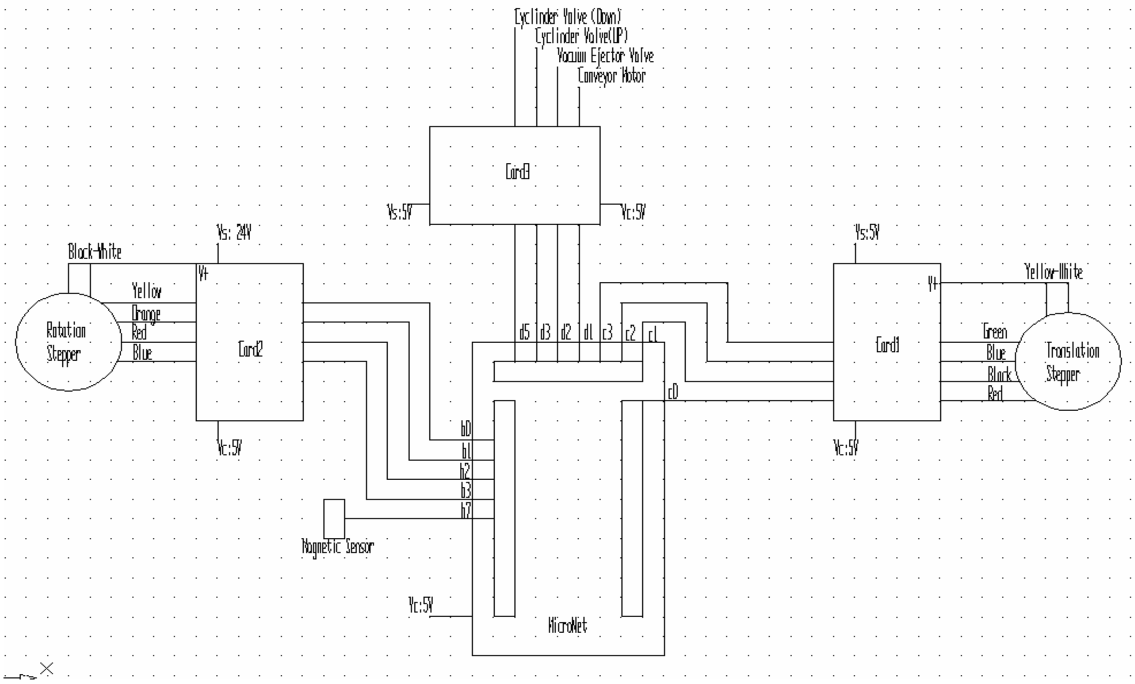


Figure 2.16. Detailed Electronic Connection Diagram.

2.1.5. Software

The software module captures and analizes the image of the moving part on the conveyor belt and controls all the system, synchronizing the manipulator movements with conveyor belt. After the captured image is preprocessed the part's orientation and position is found. When image processing operations are completed, the control phase begins. The distance of the mass center of the part from the manipulator is calculated. The number of steps is determined to reach the part. Software drives the robot manipulator to the mass center of the part to pick it up and then corrects the part's position and place it to the packaging area.

Software communicates with the microcontroller card via serial port at 9600 bps. Also robot manipulator status and the input/output control signals are monitored . The software is programmed using C language.

CHAPTER 3

COMPUTER VISION SYSTEM

3.1. Introduction

Images are vital part of our everyday life. They are used to communicate, educate, survey, etc. Human do this continuously without conscious effort. As we build machines to make life easier, we want to provide the machines with more abilities. Computer vision aims to enable machines to understand the world that they see, in real-time and without any human assistance. Similary machine vision aims provide a practical and affordable visual sense for any type of machine in the most effective way for specific applications "(Awcock and Thomas 1996)".

3.2. The Generic Machine Vision Model

The manufacturing industries commonly involve inspection and assembly tasks which are traditionally performed by human operators. Separating complex manufacturing operations into simple step by step instructions was the introduction of mechanisation. The need for systematic assembly and inspection operations being performed in sequence, or being implemented in parallel, lends itself to possible solution by vision controlled automata.

Necessary functions for a visual guided machine can be idendified "(Davies, 1997)" as ;

- The enviromental constaints
- The capture of an image
- The analysis of that image
- The recognition of certain objects with in image

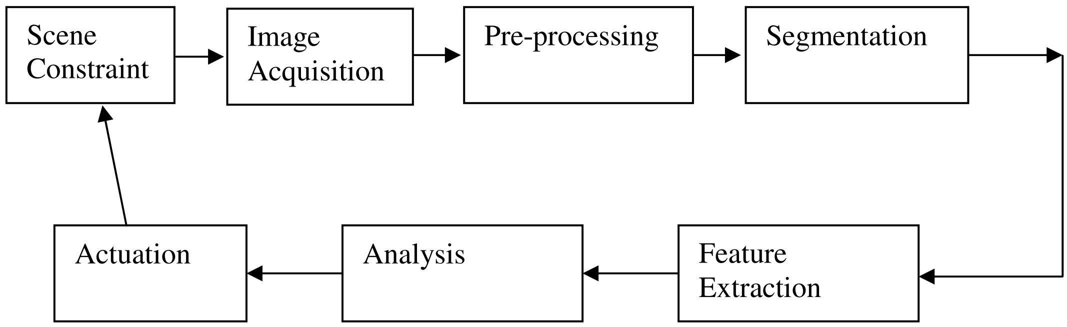


Figure 3.1. The generic machine vision model

This generic model identifies the seven sub-systems which may be identified within a typical machine vision system.

3.2.1. Scene Constraints

The scene refers to the industrial enviroment in which the manufacturing taks is to take place. The most important aim of the scene consraints sub-system is to reduce the complexity. Existing constraints may include knowledge of the limited number of objects manufactured, their appearance or contolling light effects.

3.2.2. Image Acquisition

This sub-system is concerned with the process of the transformation of optical image data into an array of numerical data which may be manipulated by a computer, so the overall aim of machine vision may be achieved.

3.2.3. Preprocessing

There are many processes in this sub-system. All the processes seek to modify and prepare the pixel values of the digitisied image to produce a form that is more suitable for subsequent operations such as segmentation. These processes can be arranged as constast enhancement and adjustment, filtering to remove noise, binarisation, correction of sensor distortion.

3.2.4. Segmentation

Segmentation is the inital step for any recognition process. Acquired image is broken up into meaningful regions and segments. In the simplest case(binary images) there are two regions: a foreground (object) region and a background region. In grey level image, thresholding methods can be classified as global or local. A global thresholding method is one that thresholds the entire image with a single threshold value. Local thresholding method one that partitions a given image into sub images, and determines threshold value for each of these sub images.

3.2.5. Feature Extraction

The feature extraction sub-system is the process of acquiring information that is necessary for interpretation. Any derived description of the objects within the image should contain all the relevant shape and size information originally contained within the stored image. The description should be invariant to position, orientation.

A number of basic parametres may be derived from an arbitrary shape, to provide valuable classification and position information. These include perimeter, center of the area and information about holes,etc.

3.2.6. Analysis

The classification sub-system is concerned with the process of pattern recognition. This process uses some or all of the object features or descriptors that have beed extracted from the image.

3.2.7. Actuation

The actuation sub-system allows interaction with the original scene. From the industrial viewpoint any machine vision system must contribute to the efficiency of production, inspection or test functions within the company. Because of this, the actuation module identifies any subseqeuent action that vision system makes, directly through robotic interactions with the original enviromental scene.

3.3. Binary Images

Binarization is a special case of intensity quantisation which leads to an image generated with only two grey levels, black(0) and white(1). Binary images are very simple to store and each pixel is represented by a single bit. They are also very easy to generate from a more general grey scale images. A threshhold value, T, is used to partition the image into pixels with just two values "(Rogers and Adams 1990)".

IF $f(x,y) \geq T$ then $g(x,y)=1$,

IF $f(x,y) \leq T$ then $g(x,y)=0$.

where $g(x,y)$ represents the binarised version of $f(x,y)$.

Binary images are very important class of images in industrial machine vision for segmentation and feature extraction.

3.4. Image Preprocessing Methods

Image preprocessing prepare the pixel values of a digitised image to produce a form that is more suitable for subsequent operations within the generic model. The two major branches of image preprocessing that are relevant in this work are image enhancement and image restoration.

Image enhancement deals with improving the quality of the image. Image restoration's aim is to recover the original image after some effects such as geometric distortion within a camera system or blur caused by movement.

In this thesis, Gaussian filter is used to remove noise in the image data. Histograms are used to find most suitable threshold values for background substraction. Thresholding is used to binarise the image. The object on the conveyor will be "0", background is "1". Preprocessing methods used in this work will be decribed in the following sections.

3.4.1. Gaussian Filter

The gaussian filter is widely used in image analysis. It enables the user to select the degree of smoothing to be achieved.

The two dimensional Gaussian Filter function can be expressed as:

$$g(x, y) = \frac{1}{\sigma\sqrt{2\Pi}}\, \varrho^{(x^2+y^2)/2\sigma^2} \quad (3.1)$$

Two dimensional filtering can be achieved by convolving the image with a one dimensional Gaussian mask in both vertical and then horizontal directions. The degree of smoothing is controlled by the shape of the Gaussian curve used,that is, it depends on the selected value of standart deviation, σ. The discrete version of the function uses a sufficent number of mask coefficients to effect the degree of accuracy required. A small value of σ effect a limited degree of smoothing while retaining fine edge detail; a large value of σ will effect greater smoothing while retaining only the major edges and lines.

These demos show the basic effects of (2D) Gaussian filter: smoothing the image and wiping off the noise. Generally speaking, for a noise-affected image, smoothing it by Gaussian function is the first thing to do before any other further processings, such as edge detection. The effect of the Gaussian function for different standard deviation values of the Gaussian filter. It can be seen in following demos.

(a) (b) (c)

Figure 3.2. Smotthing nonnoisy image (a) Lena.gif (b) Filtered with sigma = 1 (c) Filtered with sigma = 3

(a) (b) (c)

Figure 3.3. Noise Cancelling (a) Noisy Lena (b) Filtered with sigma = 1 (c) Filtered with sigma =3

3.4.2. Histogram Analysis

In image processing, the histogram of an image normally refers to a histogram of the pixel intensity values. The histogram can be observed graphically showing the number of pixels in an image at each different intensity value in image. For an 8-bit greyscale image there are 256 different possible intensities. If the x-axis is intensity value of pixel(0-255) and the y-axis is the number of the occurance of that pixel value, we can find the intensity distrubition graph (Histogram) for that image. Histograms are used in many applications. One common application is to decide what value of threshold to use when converting a greyscale(0-255) image to a binary image('0','255') by thresholding. If the image is suitable for thresholding then the histogram will have two peaks (Bimodal Histogram).

A suitable threshold for separating these two groups will be found somewhere in between the two peaks in the histogram. Mode method is one of methods that are used to determine suitable threshold value. Firstly, the histogram peaks are found and the minimum value between them is detected as a threshold value. To avoid detection of two local maxima belonging to the same global maximum, a minimum distance is usually required in gray levels between these maximas.

$$Second\ \ peak = Max\{[((k-i)^2 \times h[k])] \mid (0 <= k < 255)\} \quad (3.2)$$

Where, 'i' is the first histogram peak and 'h[k]' is the k th histogram value.

Also, smoothing techniques are usually used to prevent this problem. Bezier "(Rogers Adams 1990)" curves are generally used for smoothing histograms. Consider N+1 control points pk (k=0 to N) in 3 space. The Bezier parametric curve function is :

$$B(u) = \sum_{k=0}^{N} p_k \frac{N!}{k!(N-k)!} \; u^k \; (1-u)^{N-k} \; for \; 0 <= u <= 1 \tag{3.3}$$

B(u) is a continuous function in 3 space defining the curve with N discrete control points Pk; u=0 at the first control point (k=0) and u=1 at the last control point (k=N).

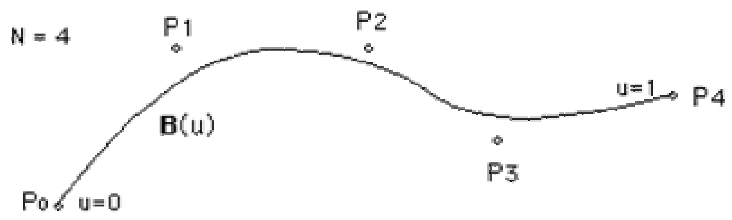


Figure 3.4. Bezier curve with 4 control points.

In this work, bezier histograms are used to determine threshold value for background substraction.

3.4.3. Thresholding

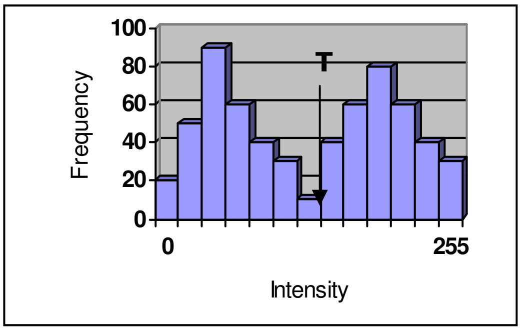
Thresholding is one of the most important techniques for segmentation and is a widely used tool for machine vision system. Binary images are much simpler to analyse than grey-scale images, but raw images often can not be converted directly to binary image. It needs some preprocessing. Therefore, there is often need to threshold a grey-scale image to obtain a binary image so that the image can be segmented into foreground and background regions. The mathematical definition of the standart binary threshold is;

IF f(x,y)>= T then g(x,y)=1,

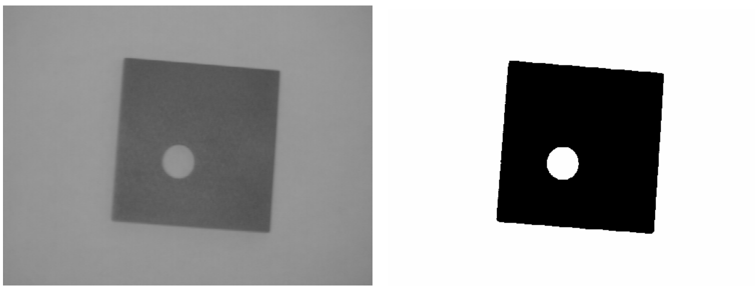
IF f(x,y)< T then g(x,y)=0,

Where T is the threshold value, g(x,y) is the binarised image of f(x,y).

Selection is the threshold value, T, is a critical issue. It is common to study the image histogram in order to find T. An image which is well suited to binarisation will feature two or more very clear peaks. The classic example is the “bimodal histogram”. In such cases the threshold value can easily be selected by finding the lowest value among the peaks.



(a) An ideal “bimodal” grey-level histogram

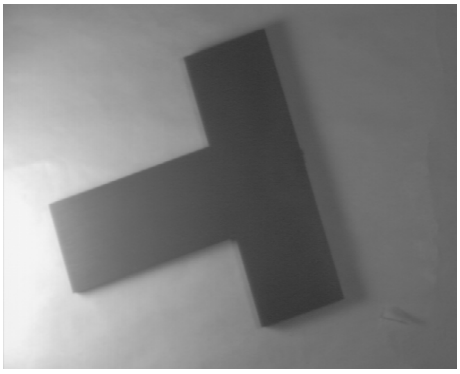


(b) (c)

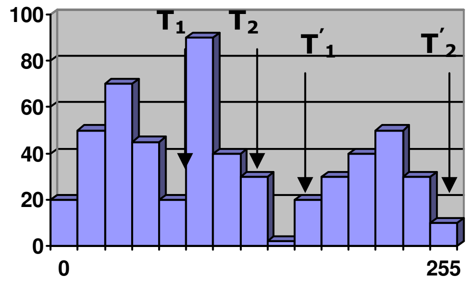
Figure 3.5. Selecting a suitable threshold value for binarisation: (a) an ideal, bimodal grey-level histogram showing the ideal threshold value, T. One peak represents the object pixels, one represents background ; (b) a typical grey- scale image; (c) the result of thresholding at T.

One variation of the simple threshold is the interval threshold operation. Here a binary output image is produced where all grey-level values falling between two threshold values T1 and T2 are converted to logic 1 and the all grey level values falling outside this interval are logic 0.

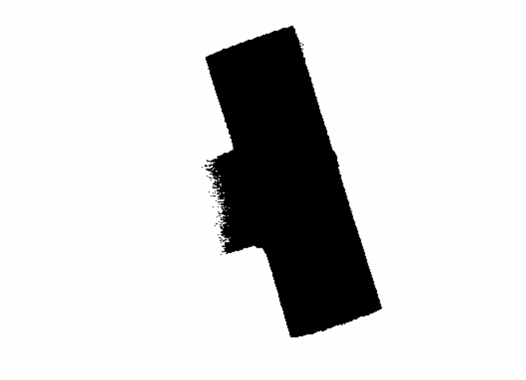
Threshold selection is a major issue. In the sample figure, it is shown the diffrent output images as to select different threshold values.



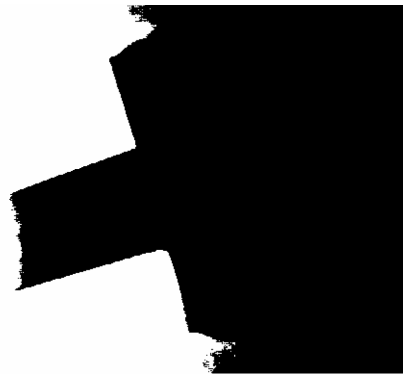
(a)



(b)



(c)



(d)

Figure 3.6. The dual threshold operation examples: (a) the original grey image; (b) its trimodal histogram; (c) the result of thresholds at T1 and T2; (d) the result of thresholds at T_1’ and T_2’.

3.5. Segmentation

Image segmentation is an important step for further image analysis. There are a many different techniques and algorithms for image segmentation. Thresholding, edge-detection, connected components labeling can be considered among the most fundamental tools for image segmentation. Connected components labeling and analysis is a grouping algorithm that has been used here to isolate, measure and identify object regions in an image. It is widely used in industrial and biomedical applications where an image often consists of objects against a contrasting background. The labeling operation assigns a unique name or number to all pixels that belong to the same connected component of the image. There are various region labeling algorithms in the literature. In this work we use recursive connected component labeling algorithm.

Recursive labeling function scans the image, finds the pixel to be labeled and labels recursively all the connected components which has the same pixel value.

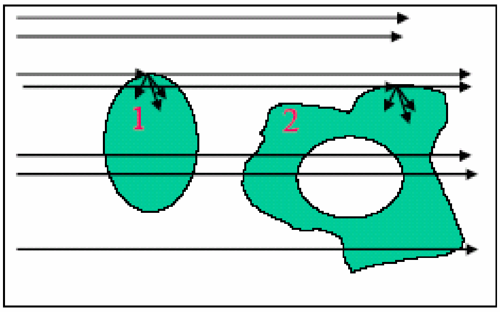
Algorithm:

-Scan the image

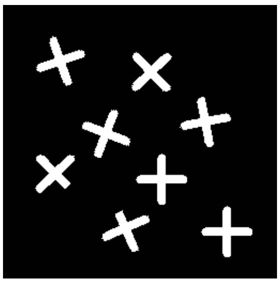
- If the current pixel value has the determined pixel value to be labeled then

- Mark current pixel as new label

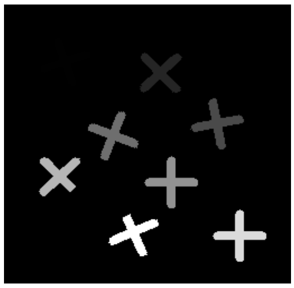
- Search for neighbour pixels recursively and mark them too.



(a)



(b)



(c)

Figure 3.7. Recursive connected component labelling. (a) Illustration of the Segmentation alghorithm. (b) Simple Binary Image. (c) Segmented (labeled) output image.

3.6. Shape Representation and Description

After an image has been segmented into regions by preprocessing operations, the segmented pixels are represented and described in a more convenient form for further computer processing operations.

Basically, an image is represented by its region (using its internal characteristics) and/or by its boundary (by using its external characteristics). This kind of representation makes the image data useful for further algorithmic analysis via a computer. The next step is to describe the image based on chosen representation. For example, an image may be represented by its boundary, and boundary is described by its features such as its length, its statistical moments, etc. These features are used to describe the object, or atributes of the object. The performance of any shape measurements depends on the quality of the original image and how well it has been pre-processed.

A boundary based representation is selected when the primary focus is on shape characteristic.

A region based representation is chosen when the primary focus is on regional properties such as color, textrures. In both representation methods, the features selected as descriptors should be translation, rotation, scale invariant.

In this thesis both regional and boundary based representation methods are used. Hereby, both representation methods will be described and discussed in following sections.

3.6.1. Regional Based Representation

3.6.1.1. Invariant Moments

The application of moments provide a method of describing the properties of an object in terms of its area,position,orientation and other precisely defined parametres. The equation describing the moment "(Hu, M. K. 1962)"of an object is given as:

$$m_{ij} = \sum_x \sum_y x^i y^j a_{xy} \tag{3.4}$$

where 'i+j' is the order of the moment,

x and y are the pixel coordinates relative to some arbitrary standart origin,
a_{xy} shows the pixel value.
Zero and first order moments is defined as:

$$M_{00} = \sum_x \sum_y a_{xy} \tag{3.5}$$

$$M_{10} = \sum_x \sum_y x a_{xy} \tag{3.6}$$

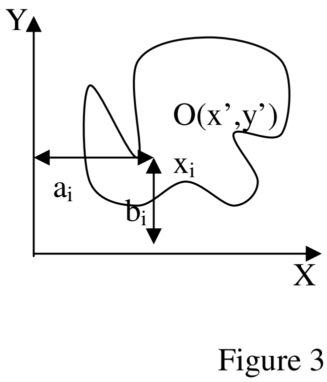
$$M_{01} = \sum_x \sum_y y a_{xy} \tag{3.7}$$

In a binary image the zero-order moment(m_{00}) is equal with the object area since a_{xy} is either '0' (Black) or '1'(White).

Any region based feature will usually require a datum point from which further features may be derived. The centroid (center of mass) is a good parameter for specifying the location of an object. Let x_1 and y_1 are the coordinates of the centroid. $(x-x_1)^2+(y-y_1)^2$ has minimum value with in the object. The centroid can also be expressed in terms of moments as:

$$x_1 = \frac{m_{10}}{m_{00}} \ \ and \ \ y_1 = \frac{m_{01}}{m_{00}} \tag{3.8}$$

where x_1 and y_1 are the coordinates of the centroid with respect to the origin.



$$x_1 = \frac{\sum_{i=1} x_i a_i}{\sum_{i=1} x_i} \tag{3.9}$$

$$y_1 = \frac{\sum_{i=1} x_i b_i}{\sum_{i=1} x_i} \tag{3.10}$$

Figure 3.8. Determining the centroid.

Moments m_{02} and m_{20} correspond to the moments of inertia of the object. It should be noted however that these basic moments have drawbacks. They vary according to position with respect to the origin and the scale and orientation of the object. A set of invariant moments would be more useful. These can be derived by first calculating the central moment, μ, with respect to the centroid, as given by:

$$\mu_{ij} = \sum_x \sum_y (x - x')^i (y - y')^j a_{xy} \tag{3.11}$$

then developing the normalised central moments, η, as:

$$\eta_{ij} = \mu_{ij}/(\mu_{00})^{\lambda} \tag{3.12}$$

where $\lambda = (i+j)/2 + 1$, and $(i+j)>2$ (first order moments are always invariant)

From these normalised moments, invariant moments, {M}, can be defined. These invariant moments are also known as Hu moments. The use of moments as invariant binary shape representations "(Freeman, H. 1961)" was first proposed by Hu in 1961 . Hu successfully used this technique to classify handwritten characters.

The set of invariant moments makes a useful feature vector for the recognition of objects regardless of position,size and orientation.

Derived Invariant Moments

$M1 = (\eta_{20} + \eta_{02})$

$M2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$

$M3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{30})^2$

$M4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$

$M5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12}) \{ (\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2 \}$
$\quad + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \{3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\}$

$M6 = (\eta_{20} - \eta_{02}) \{(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\} + 4\eta_{11}(\eta_{30} + 3\eta_{12})(\eta_{21} + \eta_{03})$

$M7 = (3\eta_{12} - \eta_{03})(\eta_{30} + \eta_{12}) \{(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\}$
$+ (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \{ 3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\}$

Orientation of an object may be defined as the angle of the axis of the minimised moment or inertia. This can be expressed as:

$$\phi = \frac{1}{2\arctan\left[2\mu_{11}/(\mu_{20}-\mu_{02})\right]} \tag{3.13}$$

where ϕ is the orientation with respect to x-axis.

3.6.1.2. Compactness

The area of a region is defined as the number of pixels in the region. The perimeter of a region is the length of its boundary. A frequent use of these two descriptors is in measuring compactness of a region, defined as $(\text{perimeter})^2$ / area. Compactness is a dimensionless quantity and thus is insensitive to uniform scale changes. Compactness is minimal for disk-shapes region. Compactness is also insensitive to orientation.

In this work compactness descriptor is used for inspecting segmented circular holes .

3.6.2. Boundary Based Representation

3.6.2.1. Chain Code

A boundary may be represented by a chain of connected steps of known direction and length. Chain code is a concise way to represent shape contour. The method was introduced by Freeman who described a method permitting the encoding of arbitrary geometric configurations. In a two dimentional orthogonal image array, there are 8 directions to move. Thus 8 compass points can be used as direction vectors. By giving a number to each direction as shown figure 3.9, the boundary of an object can be traced and coded as a squence of numbers.

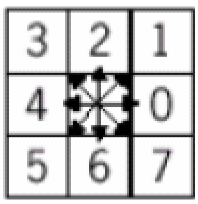


Figure 3.9. Chain Code Directions

Algorithm can be described as :

-Scan image(left to right- top to bottom)

- Pick the starting pixel

- Start tracing in clockwise direction until reaching the starting pixel.(For close contour)

e.g. 2,1,0,7,7,0,1,1,. . .

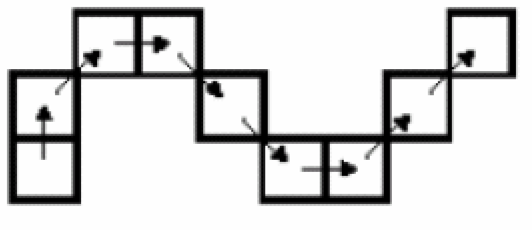


Figure 3.10 Illustration of chain code implementation

If the chain code is used for matching, it must be invariant to transformations such as translation, rotation and scaling. Invariance property can be achieved by representing the differences in chain code in the successive directions. This can be implemented by subtracting each element of the chain code from the previous one and taking the result modulo N, where N is the connectivity (i.e. 8 or 4). Resulting representation is called differential chain code representation which is translation and rotation invariant. However, this modified chain code representation is not scale invariant.

Iivarinen and Visa (Iivarinen and Visa 1996) introduced a chain code histogram (CCH) for object recognition. CCH is computed as $p(k)=n_k/n$, where n_k is the number of chain code having k value in a chain code and n is the total number of chain code elements. The CCH reflects the probabilities of different directions in a boundary. The CCH is translation and scale invariant, but it is not rotation invariant. To achieve rotation invariance, the normalized CCH is proposed. It is defined as $p(k)=l_k n_k/l$, where nk is the same as in CCH, l_k is the length of the direction k and l is the length of the contour. Thus, the normalized CCH is translation, rotation and scale invariant.

Chain code representation is sensitive to noise and variations on the object boundary. So it is not suitable to use chain code representation for this work.

3.6.2.2. Fourier Descriptors

One of the most widely used shape descriptors in shape recognition is Fourier descriptor (FD). It is used in applications such as image retrieval “(D., Lu, G. 2002)”, medical imaging, hand pose recognition “(Ellis 2004)”.

For a given shape defined by a closed curve C which in turn is represented by a one dimensional function u(t), called shape signature. At every time t, there is a complex u(t), 0<t<T, T is the period of t. Since u(t) is periodic, we have u(t+nT) = u(t). The discrete Fourier transform is given by:

$$a_n = \frac{1}{N}\sum_{t=0}^{N-1} u(t)exp(-j2\Pi nt/N) \quad n\varepsilon Z \tag{3.14}$$

The coefficients an, n = 0, 1, …, N-1, are used to derive Fourier descriptors (FDs) of the shape.

Different shape signatures (u(t)) can be used to derieve FD. Complex coordinates, Centroid distance, Curvature, and Cumulative angular function will be discussed as shape signatures.

3.6.2.2.1. Shape Signatures

Fourier descriptors are derived from a shape signature. In general, a shape signature u(k) is any 1-D function representing 2-D areas or boundaries. Different shape signatures have been used to derive FD. In the following, it is assumed that the shape boundary coordinates (x(k), y(k)), k = 0, 1, …, N-1, have been extracted in the preprocessing stage, k usually means number of boundary pixels.

3.6.2.2.2. Complex Coordinates

The boundary of a shape can be represented as a sequence of coordinates C(k) [x(k), y(k)], = for k= 0, 1, 2…N-1 where N is the boundary lenght. Each coordinate pair can be considered as a complex number such that:

$$C(k) = x(k) + jy(k) \quad (3.15)$$

The discrete Fourier transform (DFT) of C(k) is:

$$FD(u) = \frac{1}{N}\sum_{k=0}^{N-1} C(k)exp(-j2\Pi uk/N) \;\; for \;\; u = 0, 1, 2..., N-1. \quad (3.16)$$

The complex coefficients FD(u) are called the complex coordinates Fourier descriptors (CCFD).

3.6.2.2.3. Centroid Distance

The centroid distance function is expressed by the distance of the boundary points from the centroid (xc, yc) of the shape.

$$r(k) = \{[x(k) - x_c]^2 + [y(k) - y_c]^2\} \;\; for \;\; k = 0, 1, 2, ..., N-1. \quad (3.17)$$

r(k) is invariant to translation. Computation of r(k) has low cost.Figure 3.11 shows the centroid distance signatures of an apple.

Figure 3.11. The Apple Shape and Centroid Distance Signatures.

The discrete Fourier transform (DFT) of r(k) is:

$$FD(u) = \frac{1}{N}\sum_{k=0}^{N-1} r(k)exp(-j2\Pi uk/N) \;\; for \;\; u = 0, 1, 2..., N-1. \quad (3.18)$$

The coefficients FD(u) are called the centroid distance Fourier descriptors (CDFD).

3.6.2.2.4. Cumulative Angular Function

The tangent angles of the shape boundary indicate the change of angular directions of the shape boundary. The change of angular directions is important to human perception. For this reason, shape can be represented by its boundary tangent angles:

$$\theta(k) = \arctan\frac{y(k)-y(k-w)}{x(k)-x(k-w)} \tag{3.19}$$

where *w*, an integer, is a jump step used in practice. However the tangent angle function θ (*k*) can only assume values in a range of length 2π, usually in the interval of [-π, π] or [0, 2 π]. Therefore θ(k) in general contains discontinuities at end points. Because of this, a cumulative angular function is introduced to overcome this problem. The *cumulative angular function* ϕ(*k*) is the net amount of angular bend between the starting position *v*(0) and position *v*(*t*) on the shape boundary:

$$\phi(k) = \left[\theta(k) - \theta(0)\right] \pmod{2\pi} \;\; k\epsilon\left[0, L\right] \tag{3.20}$$

ϕ (*k*) is 2 π periodic, it is suitable for Fourier transform. L is the boundary length. The normalized variant of ϕ (*k*) is defined by Zahn and Roskies "(Roskies, 1972)" using normalized arclength (assuming boundary is traced counter clock-wise).

$$\psi(t) = \phi(\tfrac{L}{2\pi}t) - t \;\; where \;\; t = \tfrac{2\pi k}{L} \tag{3.21}$$

ψ(*t*) is invariant under translation, rotation and scaling. Cumulative angular signature uniquely describe a shape. However, boundary noise can cause much bigger change in the representation than the change in centroid distance therefore, the structure of ψ(*t*) is usually much more rugged than *r*(*t*) (Figure 3.12).

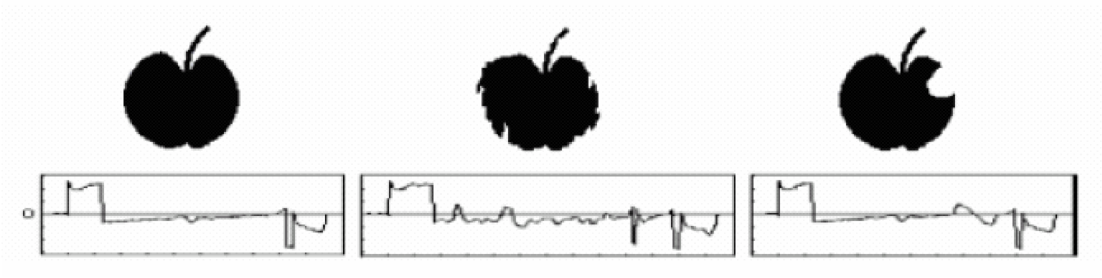


Figure 3.12 Three apple shapes and their respective Cumulative Angular Functions (Source: Zhang and Lu 2005)

The discrete Fourier transform (DFT) of (k) is:

$$FD(u) = \frac{1}{N}\sum_{k=0}^{N-1} \psi(k) exp(-j2\Pi uk/N) \;\; for \;\; u = 0, 1, 2..., N-1. \quad (3.22)$$

The coefficients FD(u) are called the cumulative angular function Fourier descriptors (CAFD).

3.6.2.2.5. Curvature Signature

Curvature is a very important boundary feature to quantify the similarity between shapes.

Curvature function is given by:

$$\kappa(k) := \theta(k) - \theta(k-1) \;\; where \;\; \theta(k) = \arctan\frac{y(k)-y(k-w)}{x(k)-x(k-w)} \quad (3.23)$$

But this curvature function has discontinuities at size of 2π in the boundary, it is convenient to use curvature function given as:

$$\theta(k) = \arctan\frac{y(k)-y(k-w)}{x(k)-x(k-w)} \quad (3.24)$$

where,

$$\phi(k) = \left[\theta(k) - \theta(0)\right] \pmod{2\pi} \;\; k\epsilon\left[0, L\right].$$

where L is the boundary lenght. Curvature is translation and rotation invariant.

The discrete Fourier transform (DFT) of κ(k) is:

$$FD(u) = \frac{1}{N}\sum_{k=0}^{N-1} \kappa(k)exp(-j2\Pi uk/N) \;\; for \;\; u = 0, 1, 2..., N-1. \tag{3.25}$$

The coefficients FD(u) are called the Curvature Fourier descriptors (CFD).

3.6.2.2.6. Invariance of Fourier Descriptors to Transformation

All the four shape signatures; complex coordinates, centroid distance, cumulative angular function, curvature signature, are invariant under translation, therefore, the corresponding FDs are also translation invariant. Rotation invariant of the FDs are achieved by ignoring the phase information and by taking only the magnitude values of the FDs.

The invariant feature vector used to index the shape is then given by:

$$f = \frac{\left|FD_2\right|}{\left|FD_1\right|}, \frac{\left|FD_3\right|}{\left|FD_1\right|}, ..., \frac{\left|FD_{N-1}\right|}{\left|FD_1\right|} \tag{3.26}$$

For centroid distance signature and curvature signature, there are only *N*/2 different frequencies in the Fourier transform, therefore, only half of the FDs is needed to index the shape. Scale invariance is then obtained by dividing the magnitude values of the first half of FDs by the DC component.

$$f = \frac{\left|FD_1\right|}{\left|FD_0\right|}, \frac{\left|FD_2\right|}{\left|FD_0\right|}, ..., \frac{\left|FD_{N/2}\right|}{\left|FD_0\right|} \tag{3.27}$$

Cumulative angular function is itself translation, rotation and scale invariant. Due to its real value, only first half of the components is necessary for the invariant feature vector of the shape. The feature vector for cumulative angular function is then given by:

$$f = [|FD_0|, |FD_1|, ..., |FD_{N/2}|] \tag{3.28}$$

All of the four FD descriptors (complex coordinates, centroid distance, curvature and cumulative angular function) are starting point invariant, because the magnitude of Fourier transform is used as FD descriptor for all of the four signatures.
Now for a model shape indexed by FD feature,

$$f_m = [f_m^1, f_m^2, ..., f_m^{N_c}] \tag{3.29}$$

and a data shape indexed by FD feature,

$$f_m = [f_d^1, f_d^2, ..., f_d^{N_c}]$$

since both features are normalized as to translation, rotation and scale, the Euclidean distance between the two feature vectors can be used as the similarity measurement,

$$d = \big(\sum_{i=0}^{N_c} |f_m^i - f_d^i|^2\big)^{\frac{1}{2}} \tag{3.30}$$

where NC is the reduced number of Fourier components needed for the feature vector.

3.6.2.2.7. Results

Chain code representation is sensitive to noise and deformations. This method is eliminated.

"Dengsheng Zhang and Guojun Lu" have made a study of Fourier descriptors generated from different shape signatures and using different Fourier invariants. The study focuses on shape retrieval application. It has been found that centroid distance r(t) is the best shape signature among the other shape signatures in terms of robustness, computation complexity, convergence speed of its Fourier series and retrieval performance of its FDs. Although cumulative angular function ψ(t) are dominantly used in deriving FDs in the literature, they show that r(t) outperforms them in shape retrieval application. Curvature is not suitable for deriving FDs due to the very slow convergence nature of its Fourier series.

According to result of "Dengsheng Zhang and Guojun Lu"'s study, we will use centroid distance r(k) as a shape signuture in this project for objects which has only one region. For more complicated scenarios other shape descriptors may be useful but in this work the representation of shape is not considered to be the primary focus of interest.

CHAPTER 4

AUTOMATIC VISION GUIDED CONTROLLER SYSTEM

4.1. System Description

The block diagram and flow chart of the entire controller system can be illustrated as in figure 4.1 and figure 4.2. The conveyor belt is activated and the system is trained for desired orientation of the objects. After training operation, the system is ready to run. The parts are captured and analysed. If the moving part is not predefined, then it is scrapted. The orientation and position are detected for predefined parts. Robot arm moves to the object position, handles the part and takes it to the rotation stepper to match the orientation. After correction, robot arm takes the part to the packaging area.

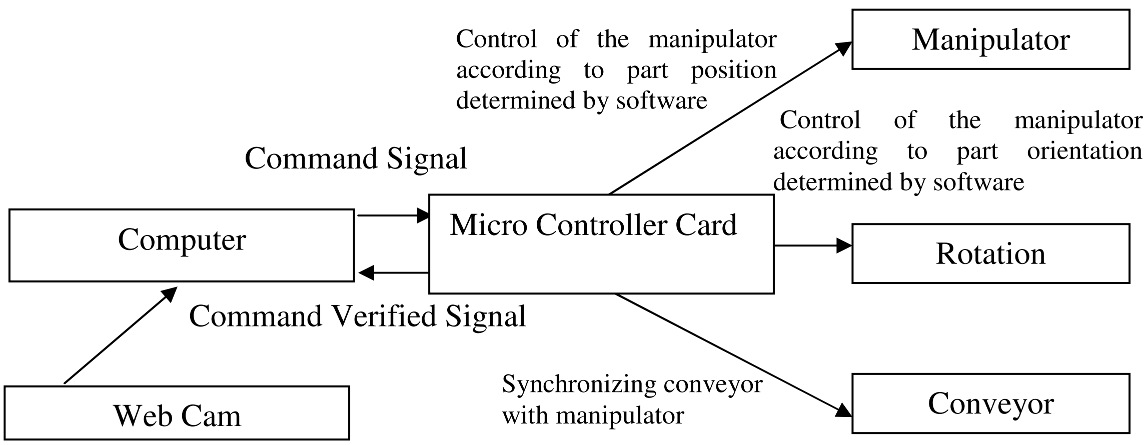


Figure 4.1 Block Diagram of the System

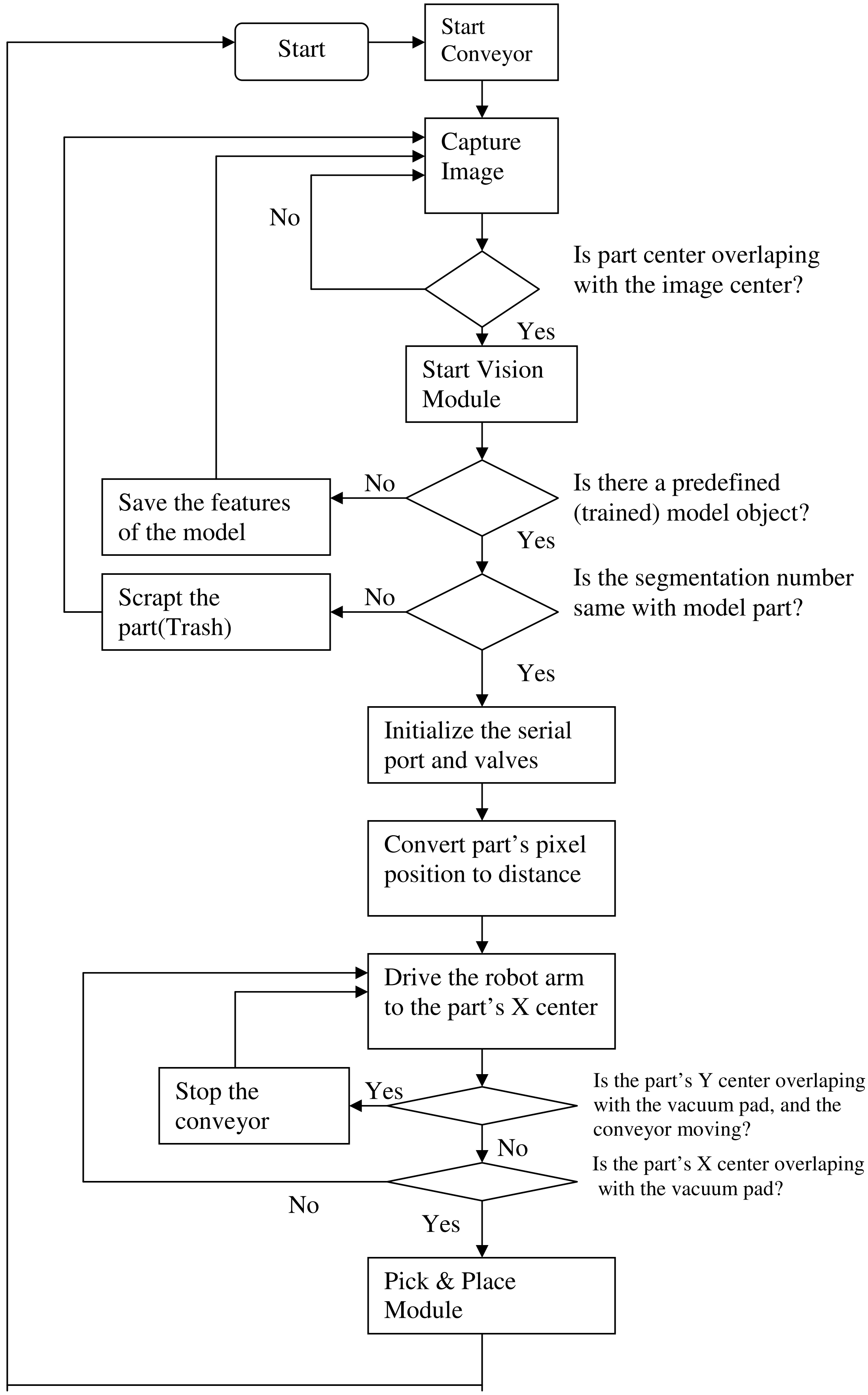


Figure 4.2 Flow Chart of the System.

There are some essential functions to be performed within the system which are explained in chapter 3. Firstly, the complexity must be reduced by constraining the scene. The parts that we used in our experiments are planar objects that contain holes which are useful image representation. There can be only one object in the system scene. The objects are fed to conveyor belt one at a time so that there are no overlapping objects. The only constraint about the colors of the object or background is that they should have contrasting colors to achieve robust background subtraction. Camera is mounted so that it is perpendicular to the conveyor-belt. These constraints are necessary to prevent distortion effects and to measure the distance between manipulator and object in pixel coordinates. Controlling the lighting effects also necessary. Conneyor-belt has a constant speed, 35mm/sec. The conveyor-belt speed should be controlled to prevent blur effects. This speed is sufficiently low to prevent blur effects.

The distances between rotation stepper, manipulator and the conveyor-belt are constant.

After these limitations to the scene, vision module procedures are applied as in Figure 4.3

Figure 4.3 Vision Module Procedures

4.2. Capturing Image Using OpenCV

The object image data is captured by OpenCV® library functions. OpenCV is an acronym for Intel Open Source Computer Vision Library. It is a collection of C functions and few C++ classes that implement some popular algorithms of Image Processing and Computer Vision. The functions used are listed below;

cvCaptureFromCAM

Initializes capturing video from camera

Function Prototype :CvCapture* cvCaptureFromCAM(int index);

index : Index of the camera to be used. If there is only one camera or it does not matter what camera to use -1 may be passed.

The function cvCaptureFromCAM allocates and initialized the CvCapture structure for reading a video stream from the camera.

cvQueryFrame

Grabs and returns a frame from camera or file.

IplImage* cvQueryFrame(CvCapture* capture);

capture : video capturing structure.

The function cvQueryFrame grabs a frame from camera or video file, decompresses and returns it.

CreateImage

Creates header and allocates data.

IplImage* cvCreateImage(CvSize size, int depth, int channels);

size :Image width and height.

Depth:

Bit depth of image elements. Can be one of:

IPL_DEPTH_8U - unsigned 8-bit integers

IPL_DEPTH_8S - signed 8-bit integers

IPL_DEPTH_16U - unsigned 16-bit integers

IPL_DEPTH_16S - signed 16-bit integers

IPL_DEPTH_32S - signed 32-bit integers

IPL_DEPTH_32F - single precision floating-point numbers

IPL_DEPTH_64F - double precision floating-point numbers

Channels: Number of channels per element(pixel),can be 1, 2, 3 or 4. The channels are interleaved, for example the usual data layout of a color image is: b0 g0 r0 b1 g1 r1 ...

The function cvCreateImage creates the header and allocates data. In this work, image bit depth is 8 and channel is 1(for gray scale image).

OpenCV uses one dimentional (1D) matrix for locating image data. It is converted two a dimentional (2D) matrix since, all the algorithms developed in this work are using 2D matrix representation for image process. The function *opencv_to_imgpro* is used for this convertion.

opencv_to_imgpro

Converts OpenCV 1D matrix to 2D matrix.

Conversion img[i] to Image[ii][jj] is ;

Img[i] , where, i= [0, img->height* img->widht] (1D);

ii=i mod (widht) ; jj= int (i/widht) .

Image[ii][jj] , where ,

ii=[0, img->widht] and jj=[0, img->height] (2D);

unsigned char ** opencv_to_imgpro(IplImage *image,int *X,int *Y);

image : OpenCV 1D image matrix.

X : Width of the image

Y : Height of the image

The function opencv_to_imgpro converts 1D image matrix to 2D image matrix and returns the new matrix.

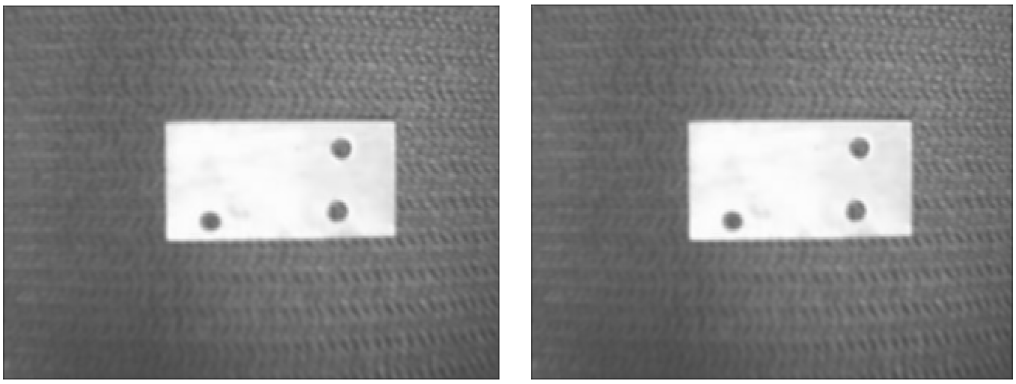
4.3. Image Preprocessing Operations

Image preprocessing operations begin when the object center is aligned with the image center.

After the image is captured and located in the 2D matrix, some preprocessing operations are needed for subsequent operations. Gaussian filtering, histogram analysis, background substraction, binarization and windowing methods are applied to the captured image in order to improve image analysis accuracy.

4.3.1. Gaussian Filtering

Gaussian filtering is the first step to be performed before any other further processings. The raw and filtered images can be seen in figure 4.4.



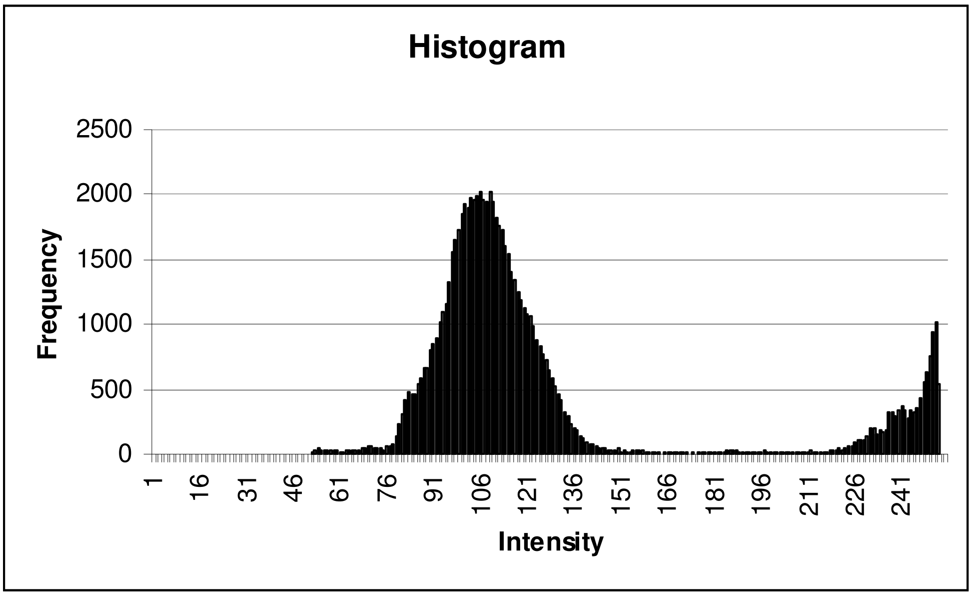
(a) (b)

Figure 4.4 Gaussian Filter : (a) Raw Image; (b) Filtered Image

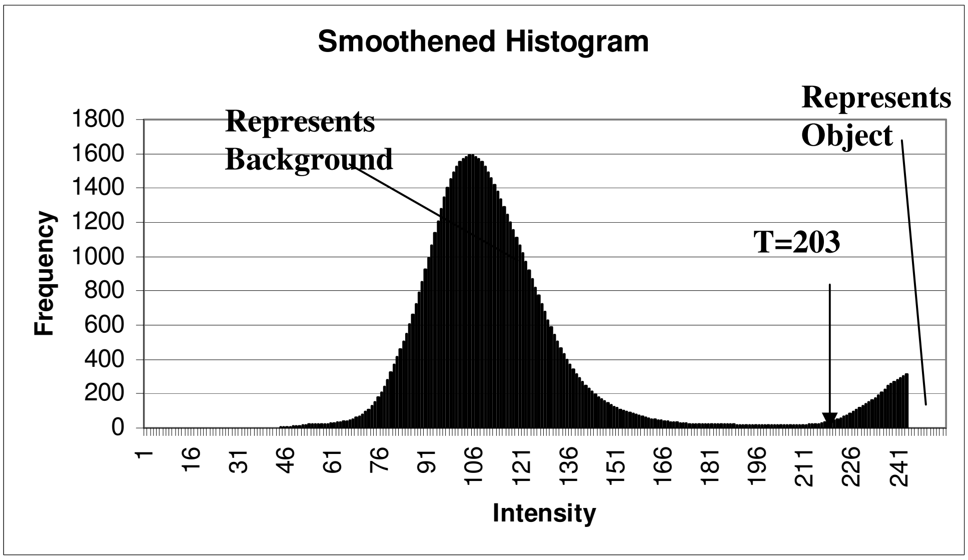
4.3.2. Histogram Analysis

Histogram from the grayscale image is extracted and smoothened by using Bezier curves. The smoothened histogram is used for finding suitable threshold values (histogram peaks) to be used in backround subtraction.

The histogram and the smoothened histogram of the captured image are in figure 4.5 . The number of the intensity values of each pixel are calculated and the most suitable threshold values are determined as explained in section 3.4.3 .



(a)



(b)

Figure 4.5. The histogram and the smoothened histogram graphics to select a suitable threshold value for the filtered image shown in figure 4.4-(b) : (a) Bimodal histogram of the image ; (b) Smoothened(Bezier) histogram.

4.3.3. Background Subtraction and Binarization

The part may contain holes, and the detection of them requires the knowledge of the background, as it becomes visible through the material. The visibility of a contrasting background through the material characterizes the presence of a hole.

Hence, the process of hole detection involves the thresholding of the image (in gray scale) by a suitable value below which the pixels represent the background region. It is assumed that the number of background pixels are more than number of part pixels.

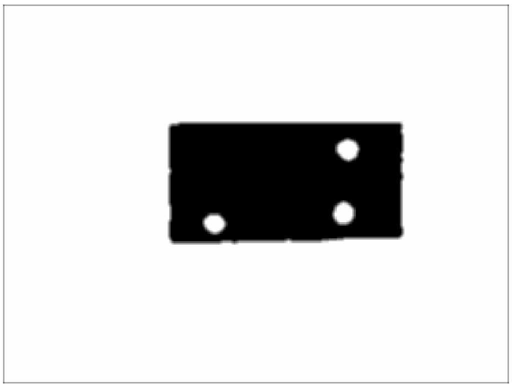


Figure 4.6 Background Substraction

4.3.4. Windowing the Object

Windowing the object is necessary for effective memory usage in segmentation operation. Investigating only the object instead of the whole image reduces processing time and cost.

Figure 4.7 Windowed Image

4.4. Segmentation Operation

The part may contain holes and they are the regions of the part. The cropped image is segmented to identify object regions and features. The Recursive Connected Component Labeling algorithm scans the image and marks the connected components with the same label number. The number of object regions may vary, therefore link lists are used to store labelled regions. Following the segmentation operation the representative features are extracted.

Figure 4.8 Segmented Image

4.5. Feature Extraction

In this thesis, both regional and boundary based representation methods are used. The number of connected components in the image determines the type of represantation. If there are more than one region in image then region based representation methods are used. On the other hand, for an image with no internal region, boundary based representation is preferred.

For each segmented region, features are stored in a link list "(Gilberg and Forouzan 1998)". In region based representation, the link list includes; labelnum, xcenter,ycenter, area, perimeter, compactness and invariant moments.

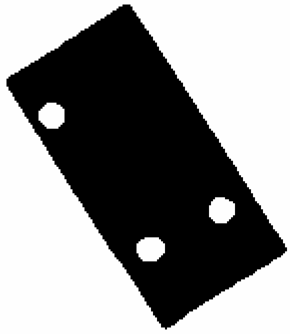
In boundary based representation, the link list stores; boundary's x and y coordinates, centroid_distances values.

The regional and boundary based information extracted from segmented images are shown in Table 4.1 and Table 4.2 .

4.5.1. Region Based Feature Extraction

A planar object is investigated as the moving part on the conveyor. The object part contains holes. Therefore, regional based methods are used to analyse position and orientation. Figure 4.9 shows binary images of model and query part.

(a)



(b)

Figure 4.9 Regional based analysis: (a) Model Image (b) Query Image

Table 4.1 Region Based Features of Model Image

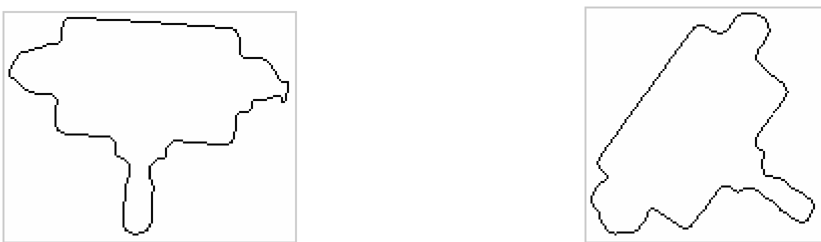
Label Number:1	**Label Number: 2**	**Label Number: 3**	**Label Number: 4**
Xcenter: 0.000000	Xcenter:40.075241	Xcenter:37.519707	Xcenter:-44.296253
Ycenter: 0.000000	Ycenter:-20.392015	Ycenter:19.932217	Ycenter:26.482088
Area: 10293.000000	Area:146.000000	Area 148.000000	Area 126.000000
Perimeter 545.000000	Perimeter:35.000000	Perimeter 36.000000	Perimeter 33.000000
Compactness:28.856991	Compactness:8.390411	Compactness:8.756757	Compactness:8.642858
0.moment : 7.358631	0.moment : 3.544564	0.moment : 3.555820	0.moment : 3.407391
1.moment : 14.268499	1.moment : 4.873681	1.moment : 5.006594	1.moment : 4.535813
2.moment : 15.644625	2.moment : 6.369622	2.moment : 5.829212	2.moment : 5.272475
3.moment : 15.676776	3.moment : 7.492049	3.moment : 7.558258	3.moment : 6.718992
4.moment : 30.451027	4.moment : 14.320668	4.moment : 14.936191	4.moment : 12.465940
5.moment : 22.649195	5.moment : 9.906848	5.moment : 10.046970	5.moment : 8.705876
6.moment : 31.343214	6.moment : 14.379551	6.moment : 14.262448	6.moment : 12.534221

Table 4.2 Region based features of query image.

Label Number: 1’	**Label Number: 2’**	**Label Number: 3’**	**Label Number: 4**
Xcenter:0.000000	Xcenter:-50.310292	Xcenter:40.306815	Xcenter:8.215503
Ycenter:0.000000	Ycenter:.-17.212124	Ycenter:15.99787571	Ycenter:40.83049446
Area:10501.000000	Area:132.000000	Area:130.000000	Area:136.000000
Perimeter:484.000000	Perimeter:34.000000	Perimeter 35.000000	Perimeter:35.000000
Compactness:22.307970	Compactness:8.626865	Compactness:9.423077	Compactness:9.007353
0.moment : 7.374348	0.moment : 3.477411	0.moment : 3.451018	0.moment : 3.481443
1.moment : 14.300188	1.moment : 4.581677	1.moment : 4.311775	1.moment : 5.225097
2.moment : 15.192292	2.moment : 5.254337	2.moment : 6.542494	2.moment : 5.821038
3.moment : 15.474687	3.moment : 6.768331	3.moment : 7.440519	3.moment : 7.129811
4.moment : 30.794840	4.moment : 12.406210	4.moment : 14.547522	4.moment : 13.687461
5.moment : 22.624375	5.moment : 8.824415	5.moment : 9.584116	5.moment : 9.418594
6.moment : 30.702705	6.moment : 12.874829	6.moment : 14.707991	6.moment : 13.463713

4.5.2. Boundary Based Feature Extraction

A pneumatic flow valve is investigated as the moving part on the conveyor. The valve contains no hole. Therefore, boundary based methods are used to analyse position and orientation of parts. Figure 4.10 shows model part and the query part’s boundries.



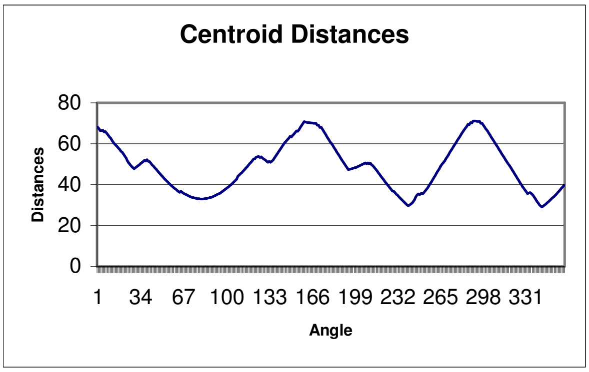
(a) (b)

Figure 4.10 Boundary based analysis: (a) Model Image (b) Query Image

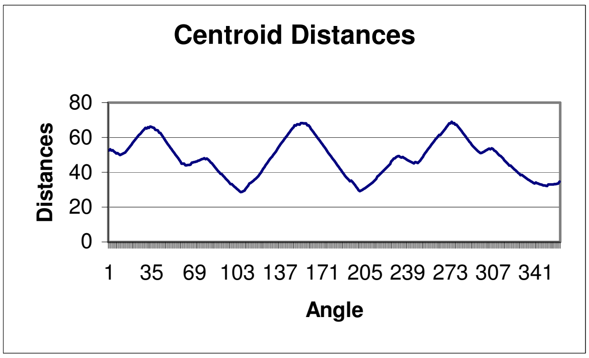
For each object, centroid distances are calculated. Both shapes are represented by centroid distances signatures. The centroid distance function is expressed by the distance of the boundary points from the centroid (xc, yc) of the shape

$$r(k) = \{[x(k) - x_c]^2 + [y(k) - y_c]^2\} \quad for \quad k = 0, 1, 2, ..., N-1 \tag{4.1}$$

r(k) is invariant to translation. Computation of r(k) has low cost. Figure 4.11 shows the centroid distance signatures of model and query shapes.



(a)



(b)

Figure 4.11. Centroid distances. (a) Model image centroid distance function ; (b) Query image centroid distance function

The analysing process begins after shapes are represented with features. The extracted features are used for image analysing process.

4.6. Image Analysis

Major focus of this work is to determine position and the orientation of parts. Image recognition can also be made by using the calculated invariant moment values for each segment. It is assumed that all the parts are the same in the line. Therefore recognition process is not necessary for this work. On the other hand, the number of holes in a part may vary. So, checking the number of the holes in a part is necessary to distinguish the images. The parts with different hole counts and coordinates are eliminated as they do not match the predefined objects. The parts with the same hole counts are processed to determine the position and orientation.

4.6.1. Determining the Position

4.6.1.1. Region Based Position Finding

For determining the position of the parts on the conveyor, the centroids should be located. Firstly, the entire object's center is calculated. The centroid (center of mass) is a good parameter for specifying the position of an object. Let Xc and Yc are the coordinates of the object center. The object center can be expressed in terms of moments as:

$$X_c = m_{10}/m_{00} \; and \; Yc = m_{01}/m_{00} \tag{4.2}$$

where Xc and Yc are the coordinates of object center with respect to the image origin (0,0). These are also the coordinates that will be used to find the object location on the conveyor belt.

The centroids of the segmented object regions are calculated according to formula 4.2 and these coordinates are recalculated with respect to entire object's center. The the entire object center becomes new origin. The centroids are invariant to translation.

After these operations, the object location (Figure 4.12) and all the centroids are determined in pixel values. (Shown in Table 4.1 and Table 4.2)

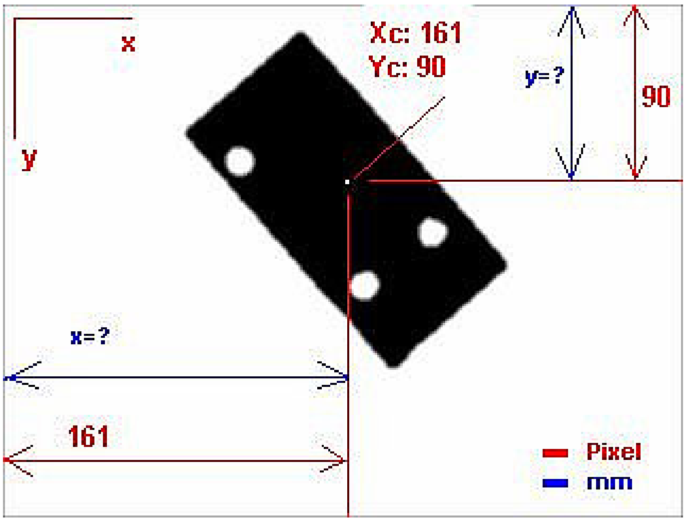


Figure 4.12 Determining the part's location on conveyor belt

After object location is determined in pixel values, the corresponding distances (in milimeters) for these pixel values are calculated as follows:

The image is 320x240 pixels. And it is also known that camera is adjusted to capture whole width of conveyor belt. The conveyor belt's width is 230 mm. So it is calculated that 1 pixel is 0.7186 mm. In figure 4.12, the part's distance to the conveyor belt's origin is x=115.69 mm.

This distance information in "mm" is converted to "number of steps" so that the stepper motor can drive the manipulator to the center of the part. The manipulator movements are explained in section 4.7.

4.6.1.2. Boundary Based Position Determining:

If the part consist of one region, the boundary based methods are applied. But to determine position of parts on coveyor, the same approach is used as in region based method. The centroid is calculated according to formula 4.2. and the location of the parts are determined on the conveyor belt.

4.6.2. Determining the Orientation

4.6.2.1. Region Based Orientation Finding

There are several methods to determine orientation of the objects. One of the methods is to investigate the angle of the axis of the minimised moment or inertia. This can be expressed as,

$$\phi = \frac{1}{2\arctan[2\mu_{11}/(\mu_{20}-\mu_{02})]} \quad (4.3)$$

where ϕ is the orientation with respect to x-axis.

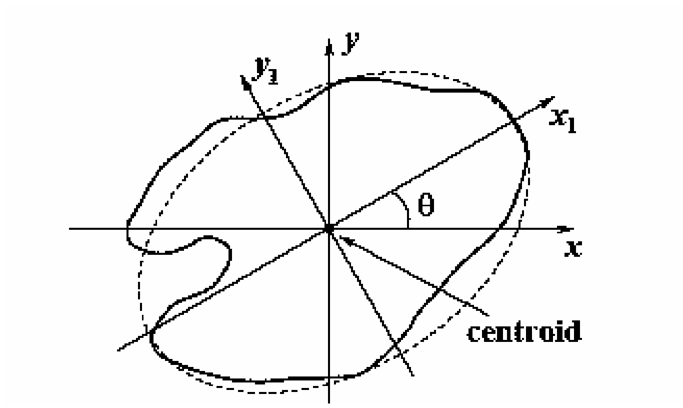
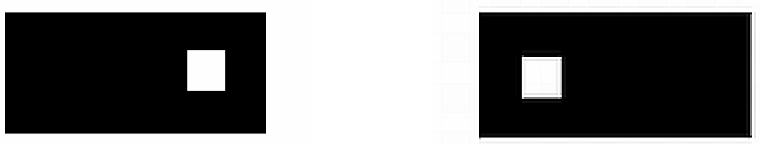


Figure 4.13 Finding orientation angle with minimised moment.

This method is not useful in this system. Because there can be a confusion as shown in figure 4.14 . Both of the images in figure 4.14(a) and figure 4.14(b) have the same orientation according to formula 4.3. But these parts do not match with each other.

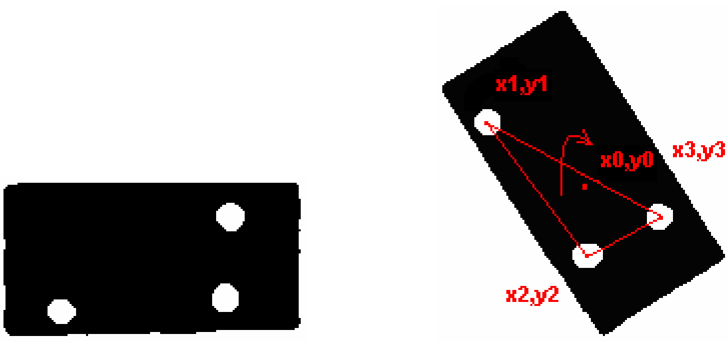


(a) (b)

Figure 4.14 : The parts have same orientation: (a) θ = 0° (b) θ = 180°

Another method is rotating the image and matching it with the model image. For each rotation angle, the differences between two image are calculated. The angle is found where the minimum euclidian distance (minimum difference) is obtained. This method is very slow and computationally very expensive. It is not efficient and useful for our system.

In this work, the parts contain holes and they can be used to find orientation of the part. Instead of rotating all the image, it is efficient to rotate just the centroids of segmented regions in the image. For each rotation angle, the differences between centroids with the model image are calculated and maximum matching is searched. The angle is found where the mimimum euclidian distance is obtained (maximum matching, minimum error).



(a) (b)

Figure 4.15. Determining the part's orientation using regional based features: (a) Model image (b) Query image

According to rotation matrix 'R', each centroid coordinates of segmented query image are recalculated for each 'θ' value between [0-2π].

$$R = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}$$

The applied rotation matrix can be shown as:

$$xrotated = x \times \cos\theta - y \times \sin\theta \quad (4.4)$$

$$yrotated = x \times \sin\theta - y \times \cos\theta \quad (4.5)$$

where xrotated" and yrotated are the centroid coordinates of query segments after rotation. The x and y are the query image's centroid coordinates ($x_1,y_1;x_2,y_2;x_3,y_3...;x_n,y_n$) and 'θ' is the rotation angle. The rotation is calculated with respect to main object's center (x_0,y_0).

For determining rotation angle 'θ' the eacludian distances between model image's centroid coordinates ($x_{m1},y_{m1}; x_{m1},y_{m1}...x_{mn},y_{mn}$) and the query image's rotated centroid coordinates are ($x_{rotated1},y_{rotated1}; x_{rotated2},y_{rotated2}, x_{rotated3},y_{rotated3}.... x_{rotatedn},y_{rotatedn}$) compared for each θ.

$$d_x = \sum_i (|xc^i_m - xc^i_{rotated}|^2)^{1/2}$$ (i=0 to number of segmented regions); (4.6)

$$d_y = \sum_i (|yc^i_m - yc^i_{rotated}|^2)^{1/2}$$ (i=0 to number of segmented regions) . (4.7)

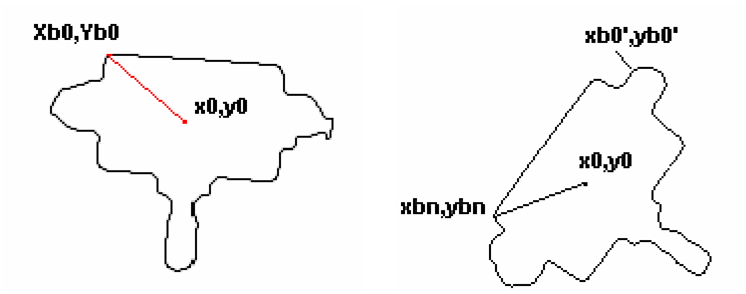
where $xc_{rotated}$ and $yc_{rotated}$ are the query image's centroid coordinates after rotation, xc_m and yc_m are the model image's centroid coordinates.

'θ' is determined where the value ($d_x + d_y$) is mimimum. After 'θ' is found, the model and query centroids of objects are sorted with respect to (Xc,Yc) coordinates and the labels are matched.

The matching results of parts Table 4.1 and Table 4.2 are ;
(label1 ↔ label1' ; label2↔ label3' ; label3↔ label4' ; label4↔ label2')

4.6.2.2. Boundary Based Orientation Determining:

Rotating the boundary pixels with respect to image center and finding maximum matching is a way for determination the part's orientation.
In this work, a different way is followed. *Centroid distances* are used for determining the part's orientation.



(a) (b)

Figure 4.16. Determining the part's orientation using boundary based features: (a) Model image. (b) Query image.

In the figure 4.16 (a), (Xbo,Ybo) are the first boundary coordinates of model image and (x0,y0) is the image center. Using these two coordinates, (Xb0,Yb0) and (x0,y0), the initial angle θ_1 is calculated according to formula ,

$$\theta_1 = \arctan[((Yb0 - y0)/(Xb0 - x0)] \tag{4.8}$$

In the figure 4.16 (b), (Xbo', Ybo') are the first boundary coordinates of query image and x0,y0 is the image center. The (xbn,ybn) are the corresponding coordinates to (Xb0,Yb0). Therefore, (xbn,ybn) should be found first. The translation value from the coordinate (Xb0,Yb0) to (xbn,ybn) has to be determined to reach (xbn,ybn). Centroid distance functions of two image are matched to find the translation value.

For translation k=0,

$C_{d0}, C_{d1}, C_{d2}, \ldots, C_{d360}$

$C'_{d0}, C'_{d1}, C'_{d2}, \ldots, C'_{d360}$

For translation k=1,

$C_{d0}, C_{d1}, C_{d2}, \ldots, C_{d360}$

$C'_{d360}, C'_{d0}, C'_{d1}, \ldots, C'_{d359}$

...

For translation value k=m,

$C_{d0}, C_{d1}, C_{d2}, \ldots, C_{d360}$

$C'_{d(m)}, C'_{d(m+1)}, C'_{d(m+2)}, \ldots, C'_{d(m-1)}$ (4.9) where "C_{d0}, C_{d1} ,C_{d2} , . . ., C_{d360}" are model image centroid distances; " $C'_{d(m)}$, $C'_{d(m+1)}$,$C'_{d(m+2)}$, . . ., $C'_{d(m-1)}$ " are query image centroid distances; k is the translation value between [0-360] and m= (0- k) mod 360.

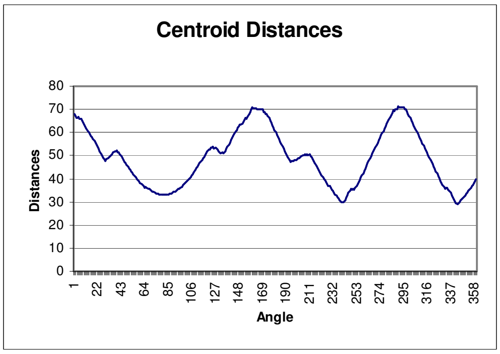
Eacludian distances are calculated for each translation value.

$$d = \sum_i (|C_d^i - C_d^m|^2)^{1/2} \ where \ \ i = 0, 1, 2, ...360; \tag{4.10}$$

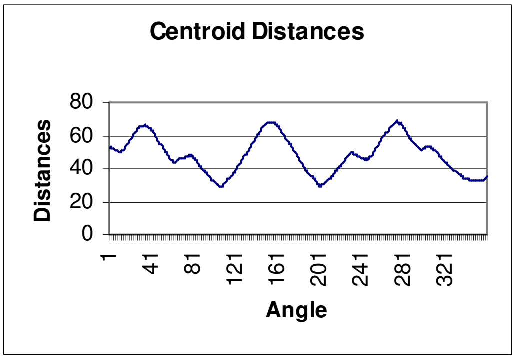
Translation value is determined where the value (d) is mimimum shown in Figure 4.17. The xbn and ybn coordinates are determined using this translation value. The second θ_2 value is calculates as,

$$\theta 2 = \arctan[((Ybn - y0)/(Xbn - x0)] \tag{4.11}$$

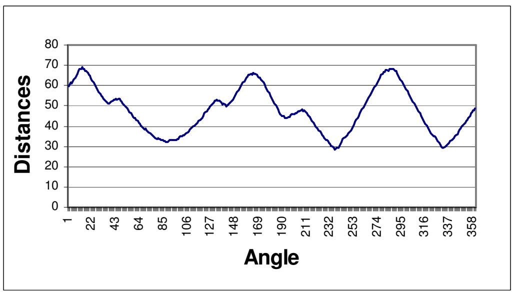
The rotation angle θ is found as $\theta=\theta_2-\theta_1$.



(a)



(b)



(c)

Figure 4.17. Determining translation value. (a) Model image centroid distances function (b) Query image centroid distances function. (c) Centroid distance function after translation the graph(b) value of k=131.

The maximum matching can be seen between two graphs in figure 4.17(a) and (c). Rotation angle is found θ=303°.

4.7. Actuation and Controlling Manipulator Movements

4.7.1. Robot Arm Movement

After the position and orientation of the part are computed, these values are used in pick and place module. The necessary number of steps are calculated to pick the part according to its distance to the manipulator. Similarly, the number of steps is calculated for the rotationing stepper according to part's orientation.

As shown figure 4.18., d1 and d2 are the known constant values and d3 is the part position on the conveyor. If the query part has the same hole count as the model part the pick and place module begins its operation. Firstly, the manipulator moves the distance d2+d3 and reaches the part's center. The system stops the conveyor belt as the part's ycenter is aligned with the manipulator axis. The time needed for alignment is calculated using the fact that conveyor belt has a constant speed v=35mm/sec. After manipulator has reached the part's center, system sends the cyclinder-down command to the controller card. The cyclinder has a magnetic sensor at its bottom as a stroke limiter. It provides the cyclinder stroke to be adjustable according to part's thickness. Cyclinder moves down until the magnetic sensor is activated. Then, vacuum pad is activated to hold the part and cyclinder picks it. There are two probability in this case. First, if the part orientation is the same as predefined, the part is taken to the packaging area directly moving the distance d1+d2+d3. If the part's orientaion is not the same as predefined, the part is taken to the rotation stepper moving the distance d2+d3; the orientaion is corrected and the part is placed to the packaging area moving the distance d1.

After these operations, manipulator goes back to its home position and waits for information about the next part.

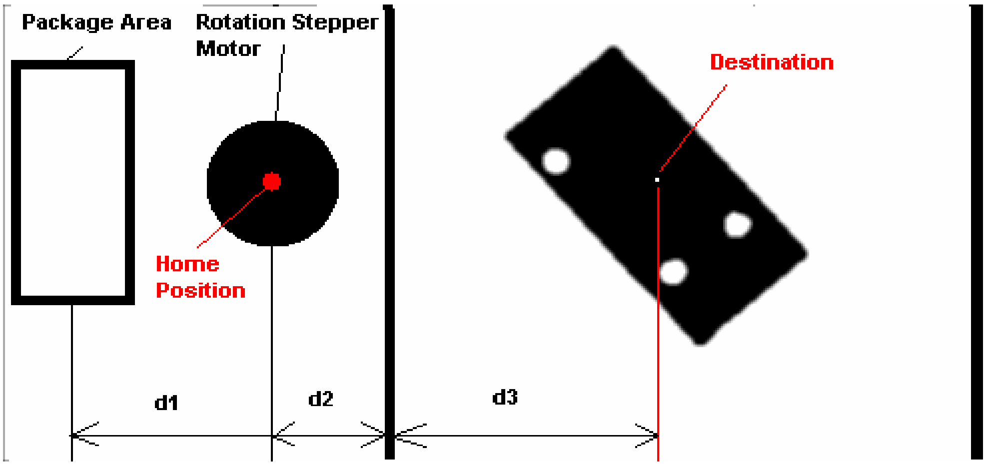


Figure 4.18 General arrangement of the system.

Translation stepper has 1.8deg/step, so 200 pulses(steps) are neccessary to achieve one revolution. The manipulator moves 50 mm/revolution that means a milimeter corresponds 4 steps.

After number of steps are determined, system sends instructions for step sequences to the controller card and controller card drives the stepper motor in full step mode. The system waits for a feedback from the controller card before sending the next step pulse. It is verified that the card has succesfuly completed the current pulse. This feed back is necessary. Otherwise, the software just assumes that stepper is ready for the next step and sends next step command, but if the stepper has not completed the current step, the step sequence fails. This is definitely something to be avoided for a stepper motor and open loop control systems. Notice that the feedback comes from the controller card, not from the stepper motor or its position. Using this feed back, the controller card guaranties that it has received the step command and performed the step pulse, but it does not have any information about the stepper's actual position. This is a open loop control system.

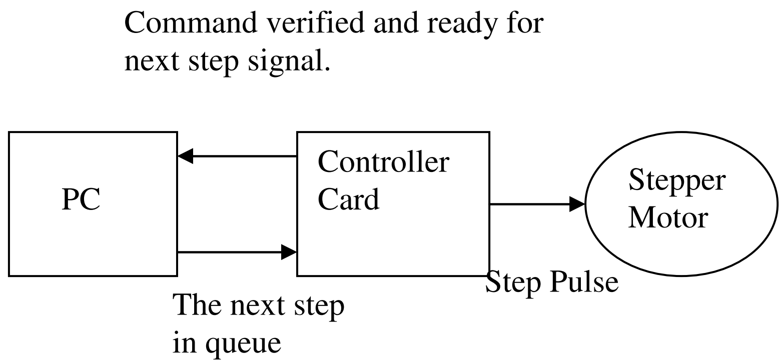


Figure 4.19 Driving the stepper motor.

The predetermined step sequences continues until the manipulator is reached to defined stations.

CHAPTER 5

CONCLUSIONS

In most of the factory automation systems, the parts are transferred by conveyor belt systems. Here, we proposed a vision guided system to control the position of parts on a conveyor. The primary advantage of vision guided conveyor applications is minimizing production cycle time, contributing to economic performance and system productivity.

The goal of this thesis was to design a vision guided manipulator prototype for use in handling and packaging automation. It is observed that the system effectively determines the position and orientation of the moving planar part on the conveyor by using a low cost web cam. The part's orientation is corrected properly after the part is picked by the manipulator from conveyor and placed on the rotator motor. Accurate positioning is obtained for manipulator with open loop control using stepper motor with a rack and pinion configration. However it is observed that the revolution speed of the stepper motor is very low (0.25Rev/Sec). The controller card's characteristics cause this problem. The controller card can process only 52 commands per second and its maximum speed is not sufficient for a smooth operation. It is clear that usage of a high speed controller card can prevent this problem.

In industrial applications, servo motors and their control systems are used to obtain precise positioning, high speed and high torque. Servo motors need close loop control. This also provides certain positioning, but increases the costs. In this work, the necessary positioning and torque values are obtained using stepper motors. Our sytem costs about 1200-1500€ in overall. If servo motors and their controllers were used, this cost would be at least 4000€. The costs and abilities of our system make this configration preferable as a prototype.

For the future work, 3D parts' position can be analised and servo vision methods can be applied for close loop control of manipulator that has multi degree of freedom.

BIBLIOGRAPHY

Acarnley P.P., 2002, “Stepping Motors: a guide to theory and practice: Fourth Edition” Institution of Electrical Engineers.

Awcock G.J. and Thomas R., 1996, “Applied Image Processing”, Int. Edition.

Bosch Rexroth, 2003, “Pneumatic catalog, Volume 1”, pp 62-63.

Corke P.I., Visual Control of Robot Manipulators by K. Hashimoto. World Scientific:, 1993.

Davies, 1997, “Machine Vision: Theory, Algorithms & Practicalities”, Academic press.

Ellis, T., 2004. “Recognizing Hand Gesture Using Fourier Descriptors” Proceedings of the 17th International Conference on Pattern Recognition. (23-26 August 2004) Vol.3, pp. 286-289.

Freeman, H. 1961. “On The Encoding of Arbitrary Geometric Configurations” IRE Trans. Electron. Comput., EC-I0 No.2 (June 1961) pp.260-268.

Galbiati L.J., 1990, “Machine Vision and Digital Image Processing Fundamentals”, pp 42-68.

Geisselmann H. et al., 1986, “Vision systems in industry: application examples”, Robotics,2,pp. 19-30.

Gilberg R.F.and Forouzan B.A.,1998 ,”Data Structures”, pp.69-73.

Hollingum J., 1984, “Machine Vision: The Eyes of Automation”, IFS, Bedford.

Hu, M. K., 1962, “Visual Pattern Recognition by Moment Invariants", IRE trans. Information Theory, Vol. 8, pp.179-187.

Lu D., G., 2002. “Enhanced Generic Fourier Descriptors for Object-based Image Retrieval” IEEE International Conference on Acoustics, Speech, and Signal Processing. (13-17 May 2002) Vol.4, pp.3668-3671.

Mannesman Rexroth, 1998, “Know-how in pneumatics”, pp 240-252.

Mitchell J.A. 1987, “Integrating vision sensors in the manufacturing process”, Proc. 3rd Int IMS '87 Conf pp.180-191 Intertec. Commun.

Plenum Press, New York, 1979, “Vision and Sensor Based Robots “, pp 3-22.

Retrieval ” IEEE International Conference on Acoustics, Speech, and Signal Processing. (13-17 May 2002) Vol.4, pp.3668-3671.

Rogers D.F. and Adams J.A., 1990, “Mathematical Elements of Computer Graphics”, pp. 295-298.

Rosen C.A., 1979, 'Machine Vision and robotics: industrial requirements', in G.G Dodd and L.Rossol, Computer.

Roskies C.T., 1972. “Fourier Descriptors for Plane Closed Curves” IEEE Trans. on Computer, Vol.21, No.3, pp. 269–281.

Simith D. and Johnson D., 1989, “An application of vision sensing to high precision part mating”, pp.36-42.

Todd J.D., 1988, “Advanced vision systems for computer-integrated manufacture”, pp 235-246.

WEB_1, 2006, Intel’s web site, 04/05/2006. http://www.intel.com/technology/computing/opencv

APPENDIX A

COMPUTER SOFTWARE

In this thesis, OpenCV library and Microsoft Visual C++ is used for coding. These source code and a text file, README, describing the source codes are given in a CD. A brief description of the algorithm is given as follows:

CONTROL.C: It is the main source code file that manages and synchronizes the peripherals. Within the code, first a window is needed to observe the data gathered from the usb cam. The camera used in the experiments is a PC-CAM. The frames have the size of 320x240 in pixels. The images are defined as;

IplImage image=cvCreateImage(cvSize(320,240),8,1);*

Here “(320, 240)” is the size of the image object, where 8 is the color depth and 1 is the number of channels (Gray format is used, hence number of channels is 1 here). To see the frames on screen a window is to be defined as;

cvNamedWindow(”CAMERA”,0);

Here ”CAMERA” is the name and label of the window object. Parameter 0 is given as a default number. In order to gather frames from camera a capture object is to be defined.

*CvCapture * capture = 0;*

and

capture =cvCaptureFromCAM(1);

is added to capture frames from the assumed camera. The following routine monitors the captured data.

cvShowImage("Camera", image);

If the part is moving on the conveyor belt and the setting is ready, the part is detected and analysed for its position. And then manipulator is activated.

VISION.H: It is the header file that includes necessary functions in generic machine model. The functions are used in the implementation of image preprocessing, segmentation, feature extraction and image analysis.

ACTUATE.H: It is the header file that includes movement functions. The functions are used in the implementation of rotation stepper movement, manipulator left-right and cyclinder up-down movements.

IMG_PRO.H: It is the header file that contains the C functions which are used in image processing procedures used in this thesis.