AUTOMATIC TRACKING OBJECT USING SIMPLE ROBOTIC ARM

SITI KHAIRANI BINTI ALIAS

BACHELOR OF ENGINEERING
UNIVERSITI MALAYSIA PAHANG

SUPERVISOR'S DECLARATION

I hereby declare that I have checked this project and in my opinion, this project is adequate in terms of scope and quality for the award of the degree of Bachelor of Electric (Electronic) Engineering.

Signature
Name of Supervisor: DR. KAMARUL HAWARI B GHAZALI
Position: LECTURER
Date:

STUDENT'S DECLARATION

I hereby declare that the work in this thesis is my own except for quotations and summaries which have been duly acknowledged. The thesis has not been accepted for any degree and is not concurrently submitted for award of other degree.

Signature
Name: SITI KHAIRANI BINTI ALIAS
ID Number: EA07034
Date:

Nothing that is worth knowing can be taught

Oscar Wilde

ACKNOWLEDGEMENT

To Dr. Kamarul Hawari Bin Ghazali, for being the best supervisor one could ever had, for making me strive for the best, for asking only the best I could be. Thank you.

To Mr Alias Bin Jamal and Mrs Zaiton Binti Yusof, for being the only set of parents I could ever wish for, for giving me love, care, understanding, money and others, thank you for everything. And for my sisters and brother, who inspire me to strive and fight to be on top, thank you.

To Mr Mohammad Khyru Bin Aris, for believing me and for being patient with me, for giving me courage, trust, love and care that no one else could ever do as perfectly as you. Thank you.

For my best friend, Vrinda Nair, thank you, for believing that I could be the greatest engineer in the world.

For my lecturers and friends, for the endless support and help, whether directly or indirectly, thank you.

A million thank you could not even cover the one I really want to thank, but the most important is I would like to thank You, Allah for being with me all the times, listening to my whines and sighs. Thank You, Allah, for giving me the sets of person that I would like to thank today. Thank You, for all the blesses You have given me.

ABSTRACT

This project is about improvement on the available technology on Closed Circuit Television Camera (CCTV). Nowadays, numerous shops and premises are equipped with CCTV. But, the CCTVs are only mounted on walls and could not detect movement that came out of its angle of surveillance. Therefore, if the CCTV camera is made movable and not static, then it could solve the problem on blind spot by making it movable according to the movement, even outside its angle of surveillance. The main objective of this project is to study the possibility to make the CCTV detectable and recognize any movements from outside its area of surveillance and to study the implementation of the CCTV on a simple robotic arm. The methods that will be used in this project are by using both software and hardware. On software, the image processing on MATLAB as well as Advantech Data Acquisition Card (DAQ) software will be used while on the hardware part, a simple robotic arm will be used and Advantech Data Acquisition Card (DAQ) is also useful for interfacing the software to the hardware. The tools that will be used is a PC, Advantech Data Acquisition Card (DAQ), a simple robotic arm and also a webcam which will represent as a CCTV in this study. The result of this project is that by using image processing on MATLAB could detect and recognize movements that are suspicious. The conclusion of this project is that this detection system could hundred percent detect and track suspicious movements in a control environment.

.

ABSTRAK

Projek ini mengenai pengubahsuaian teknologi kamera litar tertutup (CCTV) yang sedia ada. Kebelakangan ini, kebanyakkan kedai dan premis perniagaan telah dilengkapi dengan kamera litar tertutup ini. Namun begitu, disebabkan keadaan fizikalnya yang statik, kamera CCTV ini tidak dapat mengesan pergerakan yang berada di luar sudut pengawasannya. Oleh itu, jika kamera CCTV tersebut boleh dikawal gerakannya, masalah ini dapat di atasi kerana kamera tersebut boleh mengesan gerakan, walaupun di luar sudut pengawasannya. Objektif projek ini adalah untuk mengetahui kebolehan kamera CCTV untuk mengesan dan mengenali gerakan yang luar biasa dari luar sudut pengawasannya dan juga untuk menyambung kamera CCTV tersebut bersama sesuatu tangan robot. Kaedah yang digunapakai dalam projek ini ialah dengan menggunakan perisian komputer dan alatan luaran. Perisian computer yang digunakan ialah “image processing MATLAB” dan “Advantech Data Acquisition Card (DAQ)” manakala alatan yang digunakan ialah sebuah tangan robot dan “Advantech Data Acquisition Card (DAQ)” untuk menghubungkan perisian dan alatan. Alatan lain yang digunakan ialah kamera web sebagai wakil kamera CCTV dan komputer. Hasil daripada projek ini ialah perisian “Image Processing” dalam perisian MATLAB dapat mengesan dan mengenalpasti sebarang pergerakan asing. Kesimpulannya, sistem ini berjaya seratus peratus dalam mengesan dan mengenalpasti sebarang pergerakan asing.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

FKEE	Fakulti Kejuruteraan Elektrik Elektronik
FYP	Final year project
MATLAB	Math Works Software
DAQ	Data Acquisition Card
PC	Personal Computer
CCTV	Closed Circuit Television
BSA	Background Subtraction Algorithm

CHAPTER ONE

INTRODUCTION

1.1 INTRODUCTION

According to www.NationMaster.com [1], the total crime in Malaysia alone is 167,173. This also position Malaysia as the 32th country with highest crime recorded. Bearing this in mind, merchants, home securities and such are going extra miles to curb crimes in their premises. One of the methods used is by installing Closed Circuit Television or CCTV. Basically, a traditional CCTV is a small camera that is connected to a computer or a screen and uses video cameras to transmit a signal to a specific place [2]. The camera or the computer captures the video; the data is written on the tape and at the same time, shows it on screen. An officer is also needed to monitor the premises based on the screens. The data is used as after forensic tools as the computer does nothing but captures and shows the image. But as more researches have been done, the CCTV is being upgraded with more function and new technologies such as Internet technologies, database technologies, image processing technologies and telecommunication technologies.

However, there are still limitations that need to overcome. Of course, the main objective of installing CCTV is to monitor all location, but it is limited to certain areas only, leaving blind spots and chances behind. Moreover, this could lead to sabotage as unwanted person could disable or dismounted the CCTV from behind. To prevent this, the CCTV is mounted on a simple robotic arm, thus allowing it to be movable so that the officer could look at it from any angle without the need to do rounding every certain period of time.

Nevertheless, to unburden the officer more, the CCTV is equipped with image processing technologies that detect changes in the scene. There are several techniques such as feature-based object detection, template-based object detection and background subtraction or inter-frame difference-based detection. Background subtraction is the most popular method and will be used in this study. Stationary background differencing and silhouette is used to detect object of interest, such as people in this study. The images are extracted from consecutive frames a few seconds apart. After the images were detected by the computer, the camera will then be moved according to the movements and focus on the movements.

The aim of this thesis is to implement a system in MATLAB that is able to detect changes in the scene, as well as recognize the movements made by such changes. The system is built based on image processing using arithmetic and so on. The system will be tested by interfacing the system with an automatic robotic arm, which will move according to the movements or changes.

The goal is to have the system detect and recognize movements and for the robotic arm to move according to the movements. The work in this thesis therefore is an initial step and may not be implemented in real time.

The hardware used is a standard computer, regular webcam, and data acquisition card (DAQ) for interfacing, a simple robotic arm and a DC geared motor.

1.2 PROBLEM STATEMENT

Due to an increasing crime rates, an interest towards extra security has arisen. More and more has installed CCTV on their premises due to affordability. However, traditional CCTV still need an officer to screens the video. The officer still has to do usual rounding every certain period of time as the area of surveillance is limited. The officer, being human could miss a few changes in background, or being misled by the data itself. The data is recorded in tape and being used after an event has occurred thus losing its primary benefit as active, real time medium.

Therefore, there is a need to develop a system that could detect and track such suspicious movements and interfacing it with a simple robotic arm that could be moveable according to the suspicious movements itself.

1.3 OBJECTIVES

The main objectives of this study are:

I. To develop a CCTV with a system for detecting and recognizing suspicious movements on its surveillance area.
II. To develop a simple robotic arm.

1.4 SCOPES OF STUDY

The study was carried out using Closed Circuit Television as the medium to capture images, shows it on screen and stores it in tape. The image will then be processed using image processing in MATLAB using background differencing or background subtraction. The image processing technology will be able to detect movement of people or suspicious movements and will be moving according to the movement itself. To do so, the CCTV is mounted on simple robotic arm to make it movable. The study also consists of designing software to recognize movements and fabricating the model of simple robotic arm with the CCTV mounted on it.

1.5 PROCESS FLOW CHART

Figure 1.1 shows the separation of information or processes in a step-by-step flow and easy to understand diagrams showing how steps in a process fit together. This makes useful tools for communicating how processes work and for clearly due time limitation on how a particular job is done in FYP 1 and FYP 2

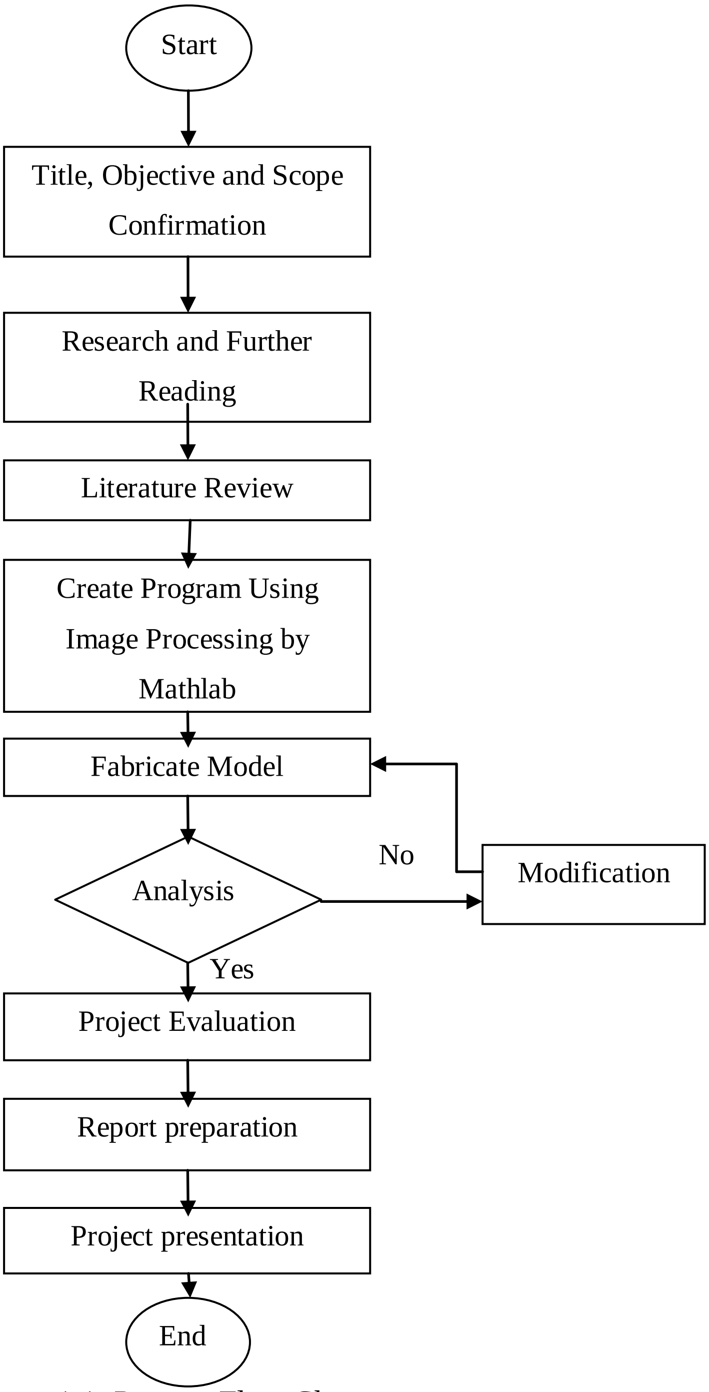


Figure 1.1: Process Flow Chart

CHAPTER TWO

LITERATURE REVIEW

2.1 INTRODUCTION

Over the past few years, the crime rates have increase dramatically. In order to curb these crimes, many premises have been equipped with CCTV for extra security. The main aim is to surveillance the whole location. These lead to the needs of an officer or more to survey the premises every certain period or so. More technologies have been added to CCTV's main function so that the improved CCTV could detect movements and so on. However as the CCTV is immobile, there are only certain angle of surveillance could be done. Therefore, the CCTV is mounted on a simple robotic arm to allow it to move accordingly.

2.2 AUTOMATIC TRACKING OBJECT USING SIMPLE ROBOTIC ARM

2.2.1 Automatic tracking object using simple robotic overview

Tracking desired object such as people have been done intensively over past year. Especially after the September 11 attack, many researcher have done studies upon automatic tracking suspicious movements in common premises such as airports, train stations, shop lots and so on. [3] People detection techniques could be taken into account to restrict the area of interest for an high-level recognition module (Schneiderman and Kanade, 2000 and Gross et al., 2002) or to identify dangerous behaviours (Collins et al., 2001). [4] For example, it can used to detect movements in restricted areas, follow the suspicious movements until safety measures have been taken. Automated video surveillance systems monitor CCTV systems and detect anomalous or suspicious behavior [5-9] with a view to alerting human operators who can take appropriate actions.

To track object, video surveillance must be used. To do so, CCTV is being used in this study. The CCTV system is basically involving a computer and a camera. The camera captures the image in the surveillance area, and the computer shows it on screen. CCTV is mainly managed from a control room where an officer or more can view the area of surveillance. It is vital for these officers to maintain awareness of the activities in these areas. However human have their limitation on maintaining focus more than fifteen minutes. Or in another case, it is too late to act when the damage has been done.

In 1942, the first CCTV installed was by Siemens AG at Test Stand VII in Peenemünde, Germany. This is done as to observe the launch of V-2 rocket. CCTV system recording was used so that the scientists could record the flight of rocket, in order to find if there is any malfunction and defect safely. In September 1968, Oleans, New York has been the first city in United States to install the CCTV in order to fight crime. And in 1960's also, the UK started installing more and more CCTV system in public places to monitor crowds during rallies and appearances of public figures. Furthermore, in UK alone, there is more CCTV alone than the people. Then, the installation of cameras became more popular, both in public spaces and retail stores, as the technology developed. Today, everywhere around the world, especially in Britain, CCTV system is used to monitor roads, premises, and banks, shopping lots, personal use, city centre, stations and airports. In 1996, government spending on CCTV technology accounted for three quarters of the crime prevention budget in the UK. [2]

In addition, UK also installed the CCTV in taxis to prevent violence towards driver, police patrol cars and even most of the car has its own CCTV. As the technology evolved, the CCTV also evolved from only monitoring the place towards many other applications. The CCTV is being updated with other technologies such telecommunication technology, as Internet technologies, database technologies, and image processing technologies. Such technologies bring out certain application to the CCTV. For example, image processing technologies could detect and track people or object movements. There is also a “talking CCTV” which allows the operator to communicate directly towards the offender. [2]

Personal use of CCTV technology has become more widespread as the technology has become much easier to acquire. Many utilize CCTV systems in their own homes to catch cheating spouses, or to monitor the care of their children in "Nannycams". Many crimes have been solved thanks to CCTV usage. Some cases of abused children were also found by CCTV.

However, there is still certain limitation of this wonder device. One of it is that the CCTV could not detect movement outside its angle of surveillance. Although mounting CCTV is cheap, the cost for the officer or officers to monitor it and monitor the place certain period of time is expensive. If only the camera could move according to the officer need, then the officer could do rounding less and if the CCTV could detect the movements outside its angle of surveillance, this will unburden the officer more. Plus, if it is compatible enough and could make it detect movements and alert the security or police, it is much better as it could eliminate the usage of officer, and be less expensive for the owner.

To achieve this goal, we have to do some alteration to the CCTV. To make it mobile, we have to attach it to a simple robotic arm. And to have it detecting and tracking movements, we have to have certain technologies equipped with is, say in this study, image processing technology.

As for image processing, it is a physical process used to convert an image signal into a physical image. The image signal can be either digital or analog. The actual output itself can be an actual physical image or the characteristics of an image.

Application could be created using image processing such as satellite imagery, wirephoto standards conversion, medical imaging, videophone, character recognition, and photo enhancement. Images could also be processed in real time which essential in this study as the detecting and tracking is done in real time, and as CCTV primary benefit as real time medium.

Generally, there are three steps on image processing. The steps taken are:

I. Import an image with an optical scanner or directly through digital photography.

II. Manipulate or analyze the image in some way. This stage can include image enhancement and data compression, or the image may be analyzed to find patterns that aren't visible by the human eye. For example, meteorologists use image processing to analyze satellite photographs. Some other techniques are enhancing images by intensity by using histogram stretching, histogram equalization, and histogram adjustment. Other than that, enhancing images using arithmetic operations such as addition which increase image brightness, multiplications to increase image sharpness, subtraction and division to detect changes in images could also be done.

III. Last be least, output the result. The result might be the image altered in some way or it might be a report based on analysis of the image. The image output could also be based on what techniques are being used.

These three steps are the basic for image processing. However it could add up according to the application preference. In real time medium, the image is fed from the camera or CCTV and was analyze according to the software. Foreground detection algorithms are normally based on background subtraction algorithms

(BSAs) [10–13], although some approaches combine this method with a temporal difference [14]. These methods are based on extracting motion information by thresholding the differences between the current image and a reference image (background) or the previous image, respectively.

BSAs are widely used because they detect not only moving objects but also stationary objects not belonging to the scene. The reference image is defined by assuming a Gaussian model for each pixel. BSAs are normally improved by means of updating their statistical description so as to deal with changing lighting conditions [9,15–16], normally linked with outdoor environments. Some authors present a different model of background, using pixels' maximum and minimum values and the maximum difference between two consecutive frames [13], a model that can clearly take advantage of the updating process. Pixels of each new frame are then classified as belonging to the background or the foreground using the standard eviation to define a threshold. After the segmentation of the foreground pixels, some processing is needed to clean noisy pixels and define foreground objects. The cleaning process usually involves 3x3 median [16] or region-based [13] filtering, although some authors perform a filtering of both images current and background-before computing the difference [12,15].

The proposed method is simpler. No model is needed for the background, just a single image. For outdoor applications this background image may be updated. Tracking algorithms establish a correspondence between the image structures of two consecutive frames. Typically the tracking process involves the matching of image features for non-rigid objects such as people, or correspondence models, widely used with rigid objects like cars. A description of different approaches can be found in Aggarwal's review, [18].

As the proposed tracking algorithm was developed for tracking people, the analysis of previous work is reduce to this particular field. Many approaches have been proposed for tracking a human body, as can be seen in some reviews [18, 19]. Some are applied in relatively controlled [12, 17, 20] or in variable outdoor [13, 16] environments. The proposed system works with blobs, defined as bounding boxes representing the foreground objects.

Tracking is performed by matching boxes from two consecutive frames. The matching process uses the information of overlapping boxes [16], color histogram back projection [21] or different blob features such as color or distance between the blobs. In some approaches all these features are used to create the so-called matching matrices [11]. In many cases, Kalman filters are used to predict the position of the blob and match it with the closest blob [20]. The use of blob trajectory [20] or blob color [16] helps to solve occlusion problems.

2.2.2 PREVIOUS RESEARCH

Below are previous studies on this matter that have been done using other techniques. It used numerous methods. The title of the study is "SecurityAgent: Security and Monitoring Computer Enhanced Video System" by P. Peer, F. Solina from Electrotechnical and Computer Science Conference ERK'OI, IEEE Region 8, Volume B, pp. 119-122, Portoroz, Slovenia, 2001. (Slovenian version). In this report, the projects that have been presented have developed in Computer Vision Laboratory which focuses on safety. The methods that have been used are based on colour segmentation and feature extraction. And it has been presented a version of automatic surveillance, using SecurityAgent system [22].

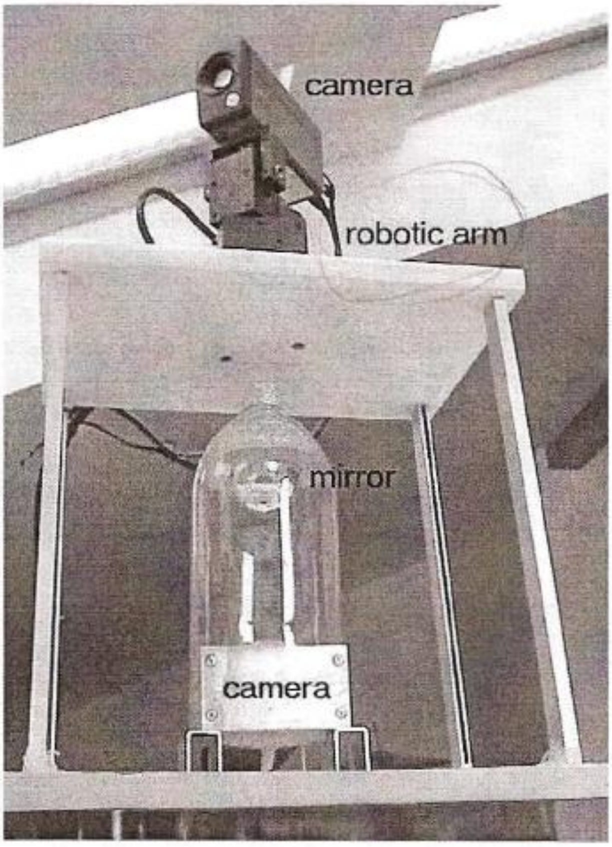


Figure 2.1: The SecurityAgent hardware

This robot consists of two cameras, a simple robotic arm, mirror and a case (figure 2.1). One of the cameras is a standard camera attached on the robotic arm and one catadioptric camera which consists of a standard camera and a spherical mirror which enables the cameras to monitor whole location in every moment.

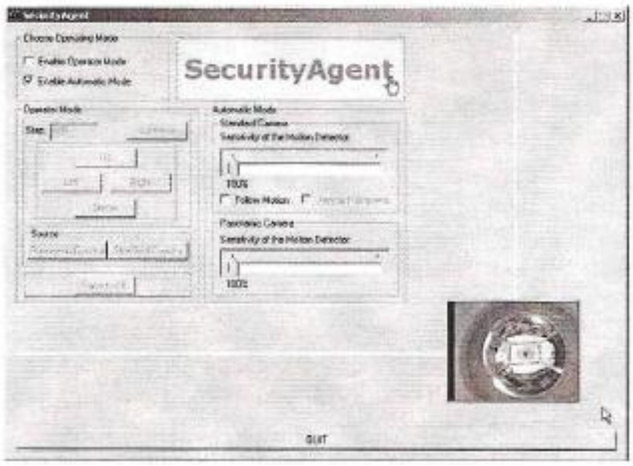


Figure 2.2: The basic user interface of the SecurityAgent

The user will be using the basic user interface of the SecurityAgent (figure 2.2). There is two mode of operation that is operator mode or automatic mode that is without the presence of an operator to monitor it. The automatic mode basic features are motion detection, motion following, making summaries of noted activities, and also notifying the operator on suspicious movements (on screen or via email). In fact, the project is now trying to notify the operator using short message system (SMS).

The title of next study is “15 seconds of fame - an interactive, computer vision based art installation” by F. Solina, P. Peer, B. Batagelj and S. Juvan which is accepted to International Conference on Control, Automation, Robotics and Vision ICARCV'02, Singapore, 2002. It is an interesting art installation which chooses any unsuspecting faces to be installed as a work of art for 15 seconds. This is inspired by Andy Warhol’s statement “in the future everybody will be famous for fifteen minutes” as well as pop art style of his famous work including Marilyn Monroe’s faces.

The digital camera lens is disguised as a precious painting (Figure 2.3), and while randomly selects the faces of candidates. The resolution of the image will be downsize to speed up the process. The system will eliminate the blobs that do not belong to face, such as too tall or too low. Then the blobs are once again gathered, and those pixels that do not belong to face, such as hands, stomach and breast are eliminated. Last but not least, the system will randomly chose one of the face candidates, and transform it using random pop-art filter which then will be aired on the monitor for 15 seconds (Figure 2.4)[23].

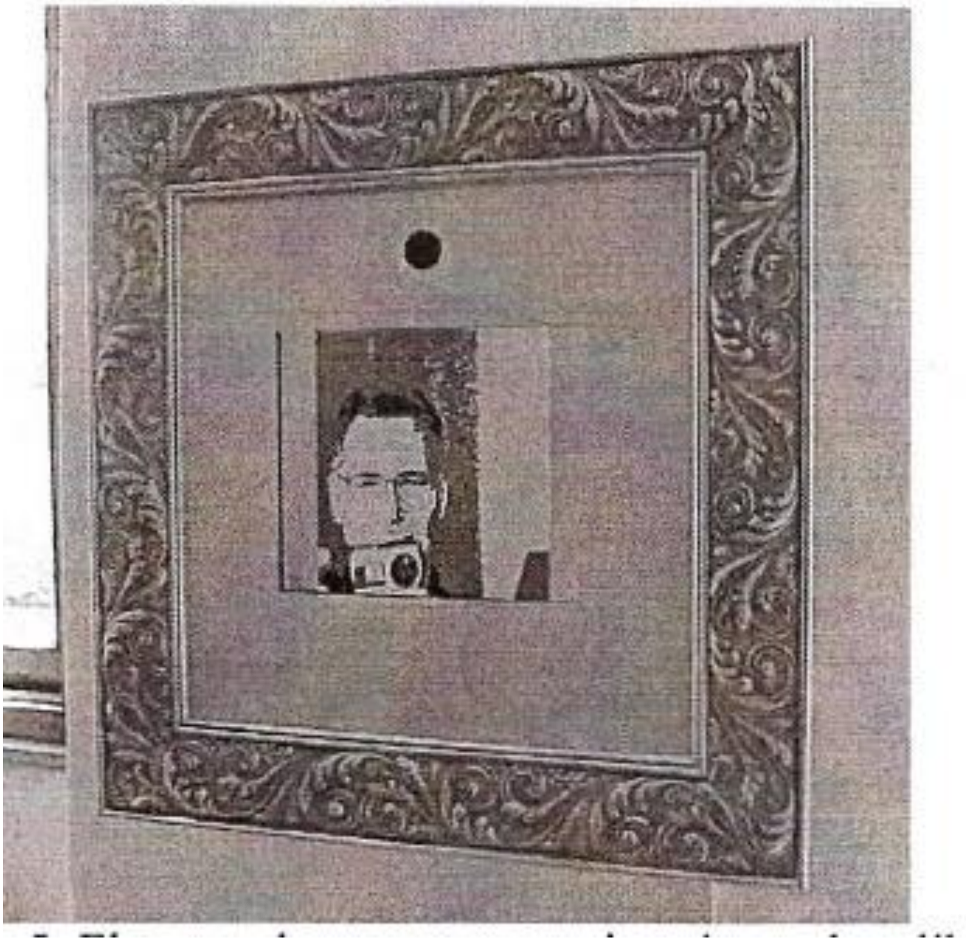


Figure 2.3**:** A flat panel computer dressed up as a precious painting for the 15 seconds of fame. The round hole on top of the painting is actually a digital camera lens.

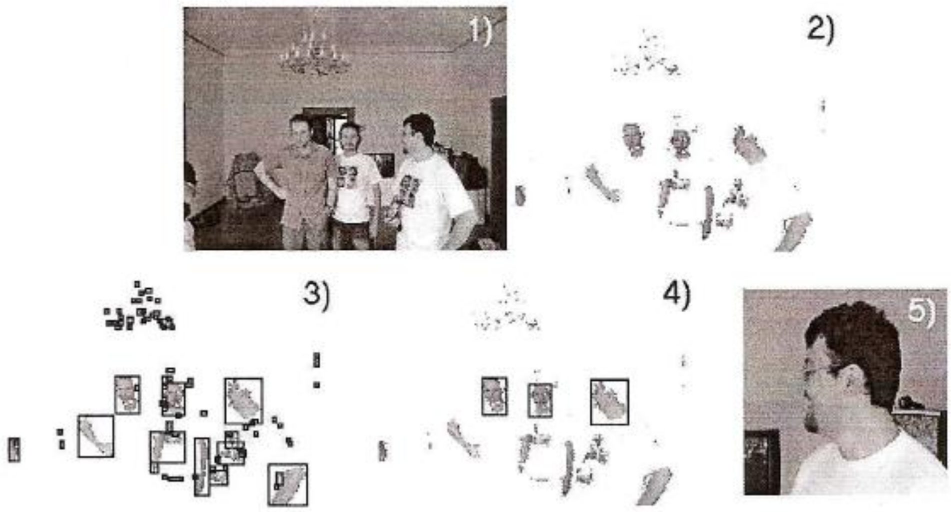


Figure 2.4: Finding face for candidate of 15 seconds of fame. 1) Downsize the resolution of the original image to speed up the process. 2) Eliminate all pixels that cannot be represented as a face. 3) Segments all the regions containing face-like pixels. 4) Eliminate all region that could not represented a face. 5) Randomly select a face and cropped it from the image. After randomly selected, the face is then transformed again using pop-art filter and will be aired on monitor for 15 seconds.

Another work by P. Peer and F. Solina, which entitle “Automatic Human Face Detection Method” on Computer Vision Winter Workshop CVWW'99, pp. 122-130, Rastenfeld, Austria, 1999. There are two distinct parts which is making face hypotheses and verification of face hypotheses. This face detection method tries to join two theories that are based on shape features, for example pair of eyes which use bottom-up features-based approach and skin colour which use colour-based approach.[24]

The basic idea of the algorithm is that it finds an image in all regions, which contain possible candidates for an eye, then on the basis of geometric face characteristics it will try to join the two candidates into a pair of an eye and finally it will confirm or refuse the face of the candidates using complexion information.(figure 2.5)

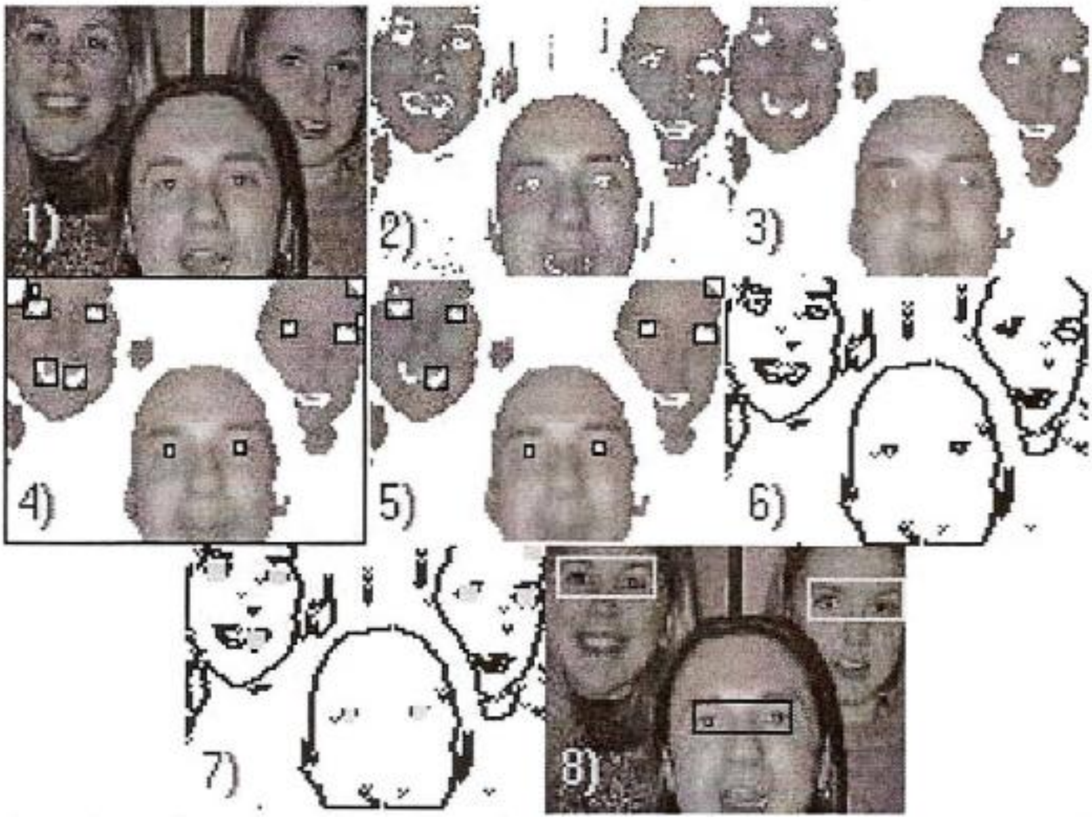
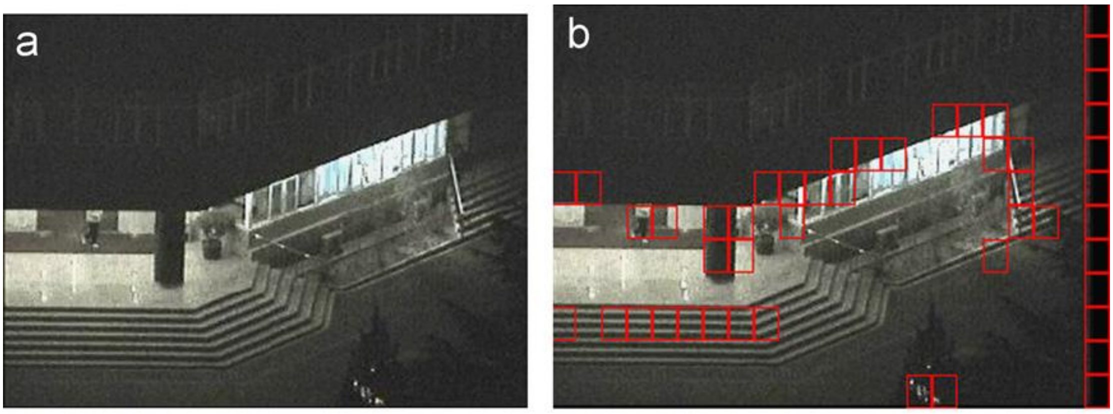


Figure 2.5: Basic principles of face detection method. 1) Input image. 2) Eliminate insignificant colour. 3) Image is filtered using median filter. 4) The white regions are segmented. 5) Insignificant region is eliminated. 6) The edges are traced. 7) The best possible circles within region of interest. 8) Output image.

A work on “a real time object detecting and tracking system for outdoor night surveillance” by Kaiqi Huang, Liang Sheng Wang, Tieniu Tan and Steve Maybank in Pattern Recognition journal 41(2008) based on detecting and tracking object at night. This is much harder as at night, without proper lighting the changes is harder to detect. However this study is based on a novel real time object detection algorithm which based on contrast analysis. At the first stage, the contrast in local change over time is used to detect potential moving object. The motion predict and spatial nearest neighbour data association are then used to suppress false alarms. [25]



c

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	1	1	0	0	1
1	1	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	1	0	0	1
0	0	0	1	1	0	1	1	0	1	0	0	0	0	0	0	0	0	1	1	0	1
0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
1	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1

Figure 2.6: Results of local contrast computation applied to a night time image. A) Original image of 320 x 240. B) A visible content of 320 x 240. C) 22 x 12 contrast salience map for visible objects

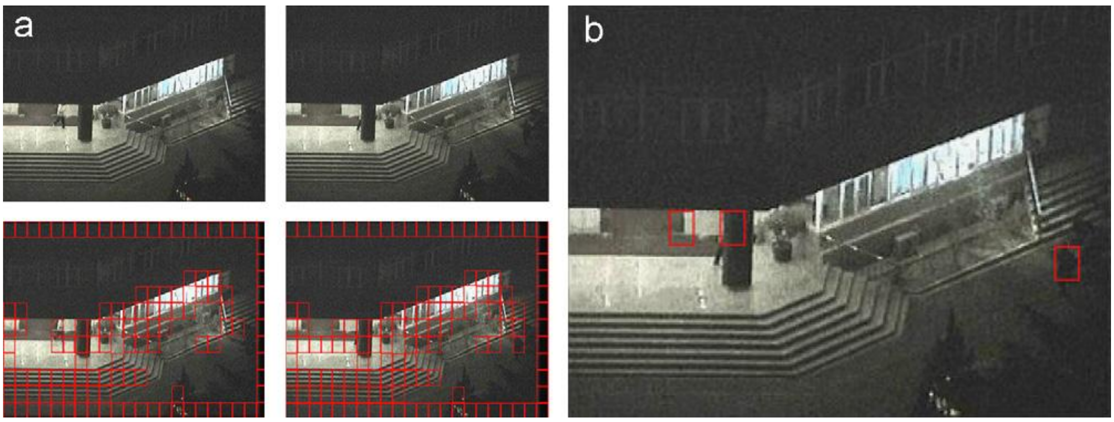


Figure 2.7: object detection. A) Object detection based on contrast. B) Moving object detection based on contrast change

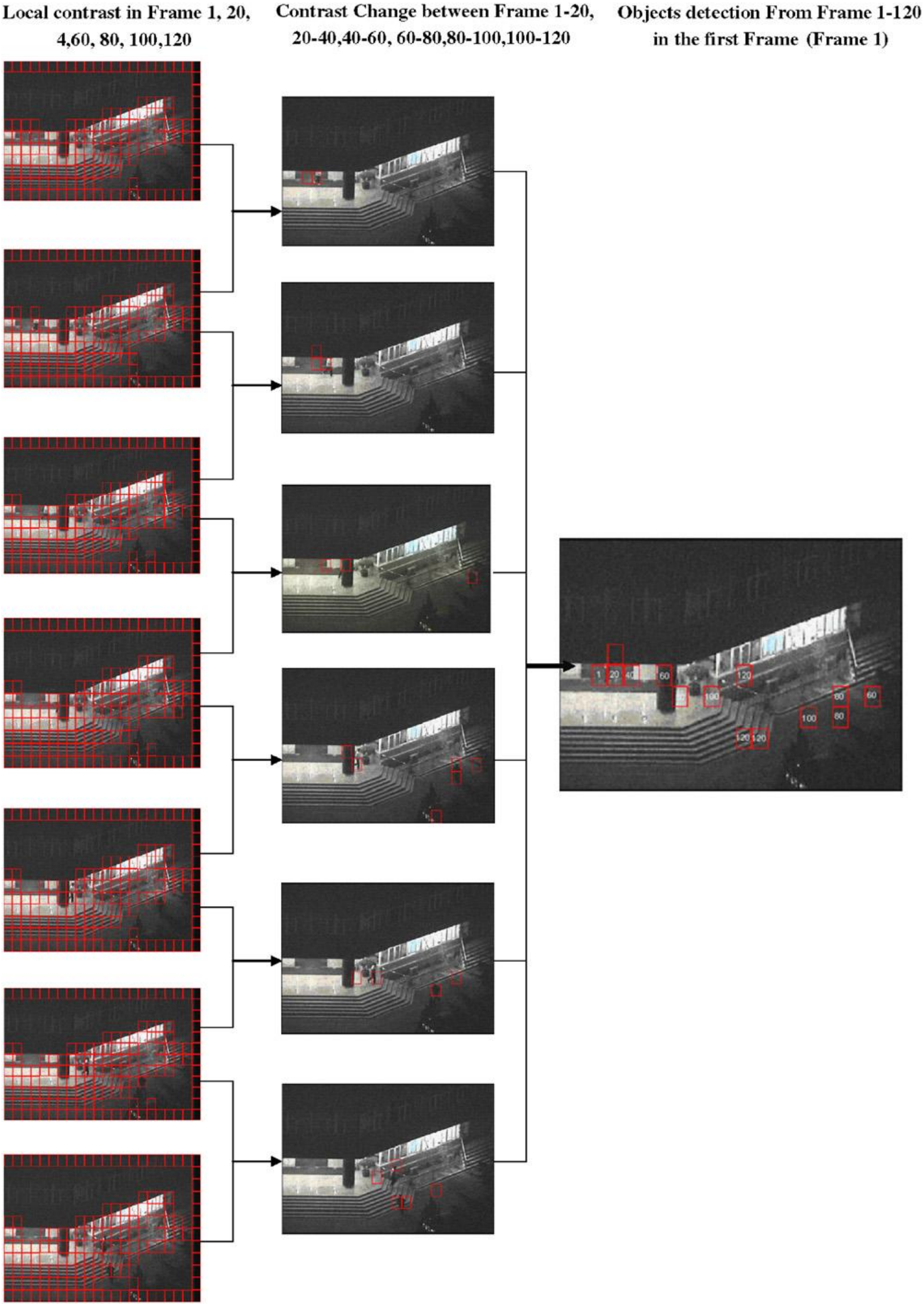
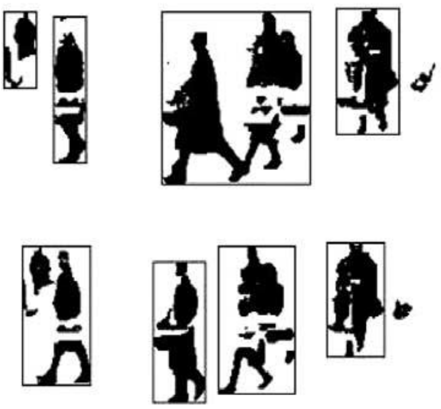


Figure 2.8: detection results for frame 1-120. First column shows visible objects found by local contrast computation. The second column show interesting objects detected by changing contrast saliency. While the third column shows consecutive detection results from frame 1 to 129 on the first frame that is frame 1.

.

Another study on algorithm that allows robust tracking of multiple objects in complex environment is by Luis M. Fuentesa and Sergio A. Valestin, entitle “people tracking in surveillance applications” that is being published in Image and Vision Computing journal 24 (2006). The foreground pixels are detected using luminance contrast and grouped into blobs. These blobs are matched from two consecutives frames, creating matching matrices that could be tracked. This method successfully solves blobs merging and splitting. [26]



$$M_{t-1}^{t} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad M_{t}^{t-1} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$S_{t-1}^{t} = \begin{bmatrix} 1 & 1 & \begin{matrix} 2 \\ 3 \end{matrix} & 4 \end{bmatrix} \quad S_{t}^{t-1} = \begin{bmatrix} \begin{matrix} 1 \\ 2 \end{matrix} & 3 & 3 & 4 \end{bmatrix}$$

Figure 2.9: an example of detected blobs in two consecutive frames, and the matching matrices and strings.

Figure 2.10: event detection, when a blob vanished where people can leave or enter the scene, the system marks the last position and raise a hiding place.

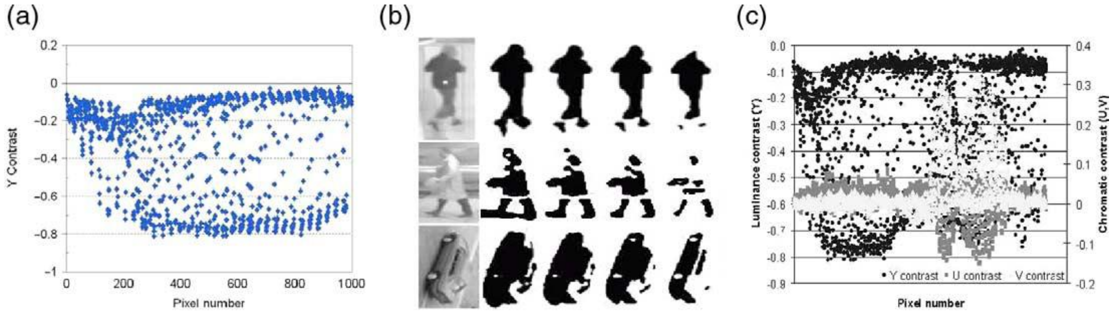


Figure 2.11**:** (a) Luminance contrast plot showing illumination variation. (b) Effect of contrast threshold values K0.10, 0.15, 0.20 and 0.40 on a foreground detection (c) color contrast plot.

Figure 2.12: four examples of foreground detection and blobs selection that is being detected by the system.

2.3 THEORY

2.3.1 Image processing in MATLAB

In electrical engineering and computer science, image processing is any form of signal for which the input is an image, such as a photograph or video frame and the output of image processing may be either an image or a set of characteristics or parameters that are related to the image. Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques to it. Image processing usually refers to digital image processing, but optical and analog image processing also are possible.

The organization of a computer vision system is highly application dependent. Some systems are stand-alone applications which solve a specific measurement or detection problem, while others constitute a sub-system of a larger design which, for example, also contains sub-systems for control of mechanical actuators, planning, information databases, man-machine interfaces, etc. The specific implementation of a computer vision system also depends on if its functionality is pre-specified or if some part of it can be learned or modified during operation. There are, however, typical functions which are found in many computer vision systems.

I. Image acquisition: A digital image is produced by one or several image sensors, which, besides various types of light-sensitive cameras, include range sensors, tomography devices, radar, ultra-sonic cameras, etc. Depending on the type of sensor, the resulting image data is an ordinary 2D image, a 3D volume, or an image sequence. The pixel values typically correspond to light intensity in one or several spectral bands (gray images or colour images), but can also be related to various physical measures, such as depth, absorption or reflectance of sonic or electromagnetic waves, or nuclear magnetic resonance.

II. Pre-processing: Before a computer vision method can be applied to image data in order to extract some specific piece of information, it is usually necessary to process the data in order to assure that it satisfies certain assumptions implied by the method. Examples are

   a. Re-sampling in order to assure that the image coordinate system is correct.
   b. Noise reduction in order to assure that sensor noise does not introduce false information.
   c. Contrast enhancement to assure that relevant information can be detected.
   d. Scale-space representation to enhance image structures at locally appropriate scales.

III. Feature extraction: Image features at various levels of complexity are extracted from the image data. Typical examples of such features are

   a. Lines, edges and ridges.
   b. Localized interest points such as corners, blobs or points.
   c. More complex features may be related to texture, shape or motion.

IV. Detection/segmentation: At some point in the processing a decision is made about which image points or regions of the image are relevant for further processing. Examples are

   a. Selection of a specific set of interest points
   b. Segmentation of one or multiple image regions which contain a specific object of interest.

V. High-level processing: At this step the input is typically a small set of data, for example a set of points or an image region which is assumed to contain a specific object. The remaining processing deals with, for example:

   a. Verification that the data satisfy model-based and application specific assumptions.
   b. Estimation of application specific parameters, such as object poses or objects size.
   c. Classifying a detected object into different categories

VI. Recognition: The classical problem in computer vision, image processing, and machine vision is that of determining whether or not the image data contains some specific object, feature, or activity. This task can normally be solved robustly and without effort by a human, but is still not satisfactorily solved in computer vision for the general case: arbitrary objects in arbitrary situations. The existing methods for dealing with this problem can at best solve it only for specific objects, such as simple geometric objects (e.g., polyhedra), human faces, printed or hand-written characters, or vehicles, and in specific situations, typically described in terms of well-defined illumination, background, and pose of the object relative to the camera.

Different varieties of the recognition problem are described in the literature

a. Object recognition: one or several pre-specified or learned objects or object classes can be recognized, usually together with their 2D positions in the image or 3D poses in the scene.
b. Identification: An individual instance of an object is recognized. Examples: identification of a specific person's face or fingerprint, or identification of a specific vehicle.
c. Detection: the image data is scanned for a specific condition. Examples: detection of possible abnormal cells or tissues in medical images or detection of a vehicle in an automatic road toll system. Detection based on relatively simple and fast computations is sometimes used for finding smaller regions of interesting image data which can be further analyzed by more computationally demanding techniques to produce a correct interpretation.

2.3.2 Robotic arm

A robotic arm is a robot manipulator, usually programmable, with similar functions to a human arm. The links of such a manipulator are connected by joints allowing either rotational motion (such as in an articulated robot) or translational (linear) displacement. The links of the manipulator can be considered to form a kinematic chain. The business end of the kinematic chain of the manipulator is called the end effectors and it is analogous to the human hand. The end effectors can be designed to perform any desired task such as welding, gripping, spinning etc., depending on the application. For example robot arms in automotive assembly perform a variety of tasks such as welding and parts rotation and placement during assembly.

In space the Space Shuttle Remote Manipulator System also known as Canadarm or SSRMS and its successor Canadarm2 are examples of multi degree of freedom robotic arms that have been used to perform a variety of tasks such as inspections of the Space Shuttle using a specially deployed boom with cameras and sensors attached at the end effectors and satellite deployment and retrieval manoeuvres from the cargo bay of the Space Shuttle.

The robot arms can be autonomous or controlled manually and can be used to perform a variety of tasks with great accuracy.

The robotic arm can be fixed or mobile (i.e. wheeled) and can be designed for industrial or home applications.

2.3.2 Interfacing using Data Acquisition card (DAQ)

Data acquisition (abbreviated DAQ) is the process of sampling of real world physical conditions and conversion of the resulting samples into digital numeric values that can be manipulated by a computer. Data acquisition and data acquisition systems (abbreviated with the acronym DAS) typically involves the conversion of analog waveforms into digital values for processing. The components of data acquisition systems include:

I. Sensors that convert physical parameters to electrical signals.
II. Signal conditioning circuitry to convert sensor signals into a form that can be converted to digital values.
III. Analog-to-digital converters, which convert conditioned sensor signals to digital values.

Data acquisition applications are controlled by software programs developed using various general purpose programming languages such as BASIC, C, Fortran, Java, Lisp, Pascal. COMEDI is an open source API (application program Interface) used by applications to access and controls the data acquisition hardware. Using COMEDI allows the same programs to run on different operating systems, like Linux and Windows.

Specialized software tools used for building large scale data acquisition systems include EPICS. Graphical programming environments include ladder logic, Visual C++, Visual Basic, MATLAB and LabVIEW.

DAQ hardware is what usually interfaces between the signal and a PC. It could be in the form of modules that can be connected to the computer's ports (parallel, serial, USB, etc.) or cards connected to slots (S-100 bus, AppleBus, ISA, MCA, PCI, PCI-E, etc.) in the mother board. Usually the space on the back of a PCI card is too small for all the connections needed, so an external breakout box is required. The cable between this box and the PC can be expensive due to the many wires, and the required shielding.

DAQ cards often contain multiple components (multiplexer, ADC, DAC, TTL-IO, high speed timers, RAM). These are accessible via a bus by a microcontroller, which can run small programs. A controller is more flexible than a hard wired logic, yet cheaper than a CPU so that it is alright to block it with simple polling loops. For example: Waiting for a trigger, starting the ADC, looking up the time, waiting for the ADC to finish, move value to RAM, switch multiplexer, get TTL input, let DAC proceed with voltage ramp. Many times reconfigurable logic is used to achieve high speed for specific tasks and Digital signal processors are used after the data has been acquired to obtain some results. The fixed connection with the PC allows for comfortable compilation and debugging. Using an external housing a modular design with slots in a bus can grow with the needs of the user.

Not all DAQ hardware has to run permanently connected to a PC, for example intelligent stand-alone loggers and oscilloscopes, which can be operated from a PC, yet they can operate completely independent of the PC.

DAQ software is needed in order for the DAQ hardware to work with a PC. The device driver performs low-level register writes and reads on the hardware, while exposing a standard API for developing user applications. A standard API such as COMEDI allows the same user applications to run on different operating systems, e.g. a user application that runs on Windows will also run on Linux and BSD.

CHAPTER THREE

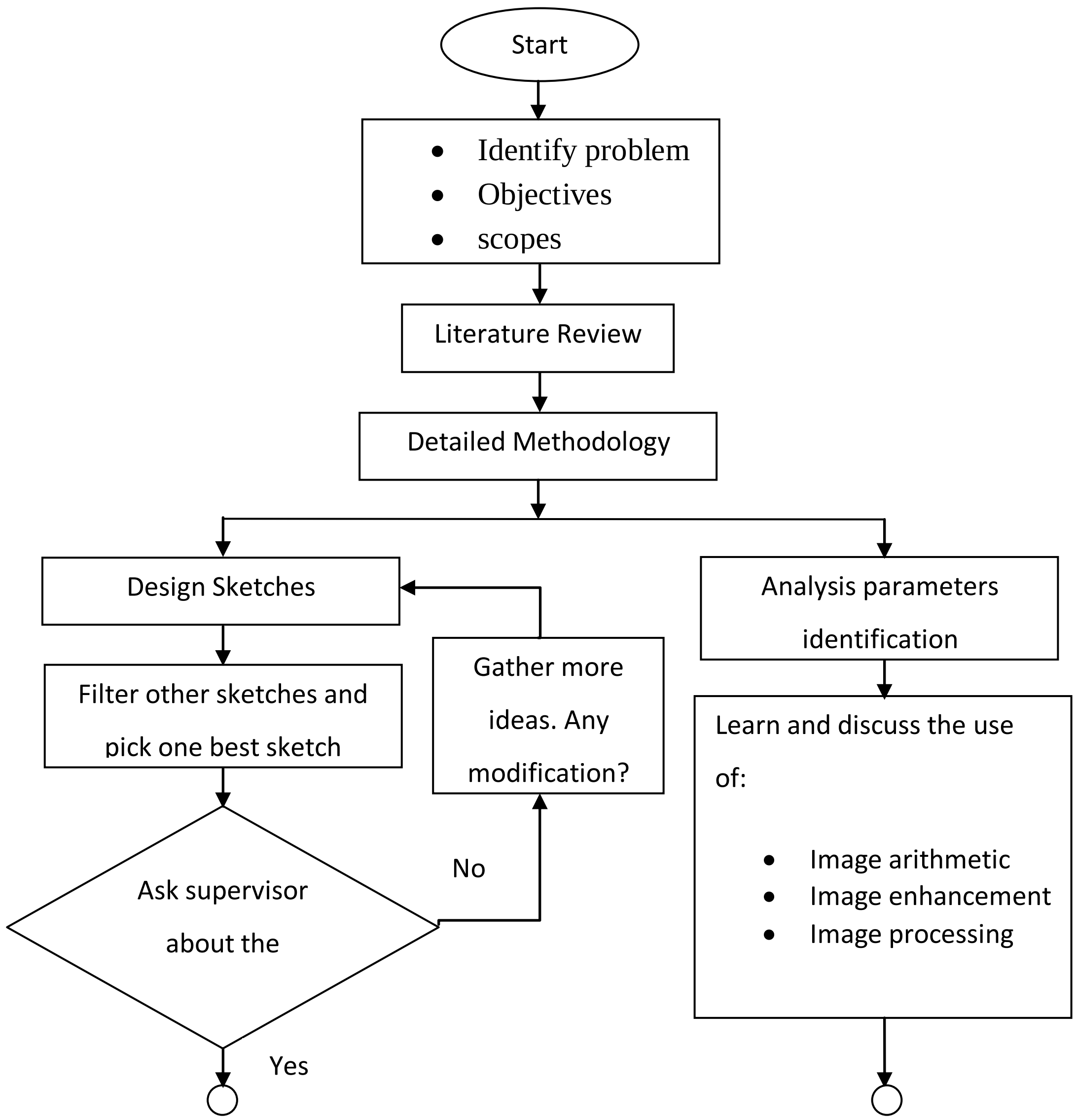
METHODOLOGY

3.1 INTRODUCTION

Methodology could be describe as a framework of process that is used to structure, plan and control the process of developing a system. Therefore, this chapter will focus on the methodology of this project. It will touch on the process flow, the hardware part as well as the software that will be used in this study.

3.2 PROJECT PROCESS FLOW

This flow chart will display how the project will progress from the research up to the analysis which is the main objective to be accomplished in this project.


Start
Identify problem
Objectives
scopes
Literature Review
Detailed Methodology
Design Sketches
Filter other sketches and pick one best sketch
Ask supervisor about the
No
Gather more ideas. Any modification?
Yes
Analysis parameters identification
Learn and discuss the use of:
Image arithmetic
Image enhancement
Image processing

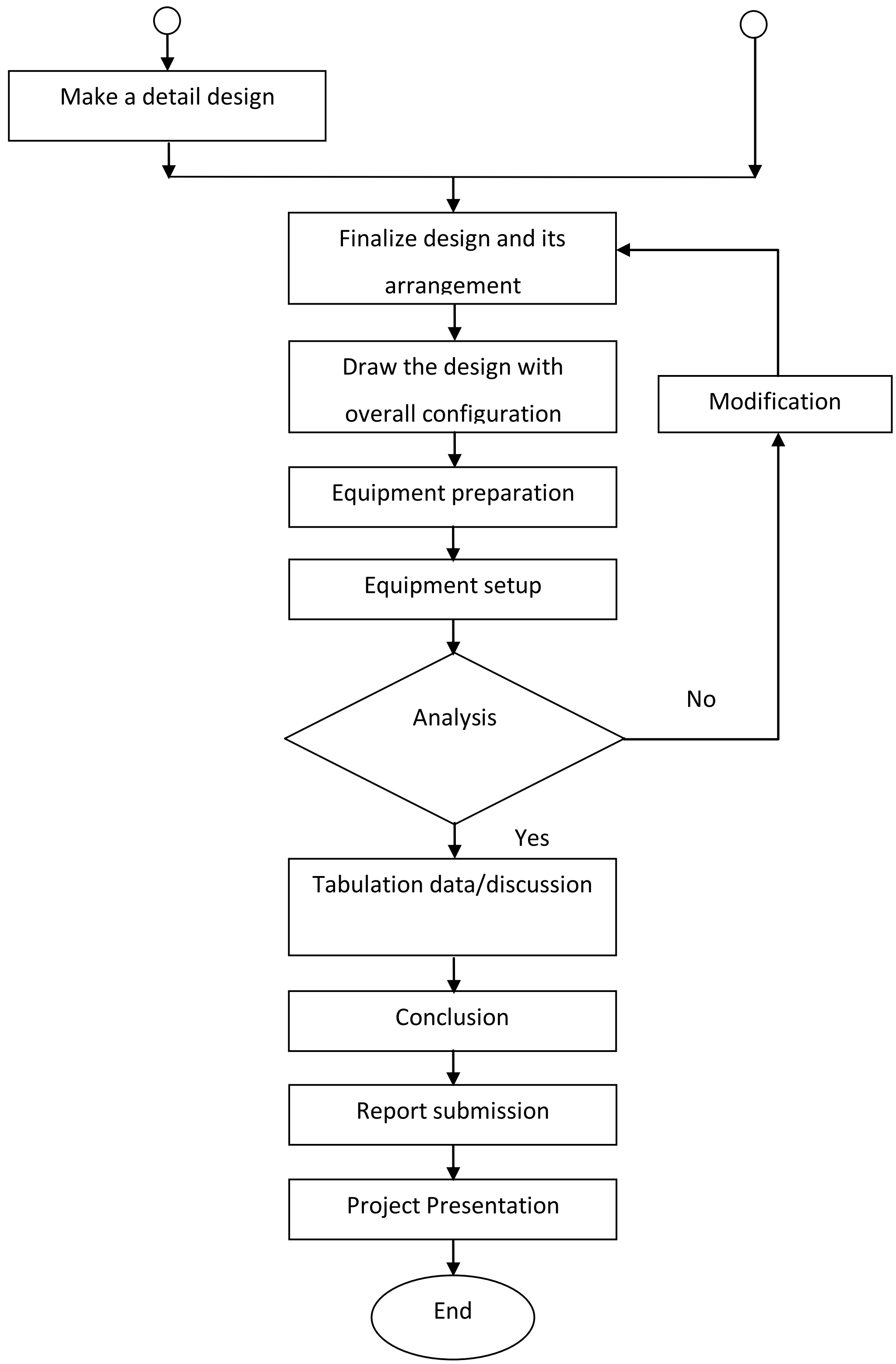


Figure 3.1: Flow chart

3.3 HARDWARE ANALYSIS

The hardware consist of three parts, that is a Personal Computer (PC), a Data Acquisition Card (DAQ), a simple robotic arm that is equipped with direct current (DC) geared motor, and a standard webcam.

3.3.1 Personal computer

The computer that is being used in this project is a personal computer with a desktop and monitor as using laptop will make the result hanging and lagging. The computer is very important as the system depends on it. It also used to connect between the software and the hardware.

3.3.2 Data Acquisition Card (DAQ)

Data acquisition is the sampling of the real world in generating data that can be manipulated by a computer. Data acquisition typically involves acquisition of signals or waveforms then and processing the signals to obtain desired information. Components of data acquisition systems include sensors that convert any measurement parameter to an electrical signal, which is acquired by data acquisition hardware.

The acquired data from the data acquisition hardware are displayed, analyzed, and stored on a computer, either using software, or custom displays and control developed using programming languages such as BASIC, C, Fortran, Java, Lisp, Pascal. Programming languages that used for data acquisition include, EPICS, Lab VIEW, and MATLAB provides a programming language but also built-in graphical tools and libraries for data acquisition and analysis.

Transducer is a device that converts physical property or phenomenon into corresponding measurable electrical signal, such as voltage and current. The data acquisition system ability to measure different phenomena depends on the transducers to convert the physical phenomena into a signal measurable by the data acquisition hardware. There are specific transducers for many different applications, such as measuring temperature, pressure, or fluid flow. DAQ also deploy various Signal Conditioning techniques to adequately modify various different electrical signals into voltage that can then be digitized using ADCs.

Signals may be digital or analog depending on the transducer used. Signal conditioning may be necessary if the signal from the transducer is not suitable for the DAQ hardware that'll be used. The signal may be amplified or deamplified, or may require filtering, or a lock-in amplifier is included to perform demodulation. Various other examples of signal conditioning might be bridge completion, providing current or voltage excitation to the sensor, isolation, linearization, etc.

DAQ hardware is what usually interfaces between the signal and a PC. It could be in the form of modules that can be connected to the computer's ports (parallel, serial, USB, etc...) or cards connected to slots (PCI, ISA) in the mother board. Due to the space on the back of a PCI card is too small for all the connections needed, an external breakout box is required. DAQ-cards often contain multiple components (multiplexer, ADC, DAC, TTL-IO, high speed timers, RAM). These are accessible via a bus by a micro controller, which can run small programs. The controller is more flexible than a hard wired logic, yet cheaper than a CPU so that it is alright to block it with simple polling loops.

Driver software that usually comes with the DAQ hardware or from other vendors, allows the operating system to recognize the DAQ hardware and programs to access the signals being read by the DAQ hardware. A good driver offers high and low level access. So one would start out with the high level solutions offered and improves down to assembly instructions in time critical or exotic applications

3.3.2.1 PCI-1710HG

The Advantech PCI-1710HG is a powerful data acquisition (DAS) card for the PCI bus. It features a unique circuit design and complete functions for data acquisition and control, including A/D conversion, D/A conversion, digital input, digital output, and counter/timer. The Advantech PCI-1710HG provides users with the most requested measurement and control functions as below:

- PCI-bus mastering for data transfer
- 16-channel Single-Ended or 8 differential A/D Input
- 12-bit A/D conversion with up to 100 kHz sampling rate
- Programmable gain for each input channel
- On board samples FIFO buffer (4096 samples)
- 2-channel D/A Output
- 16-channel Digital Input
- 16-channel Digital Output
- Programmable Counter/Timer
- Automatic Channel/Gain Scanning
- Board ID

3.3.2.2 Installation Guide

Before installation the PCI-1710HG card, make sure the following necessary component is present:

- **PCI-1710HG Multifunction card**
- **PCI-1710HG User's Manual**
- **Driver software Advantech DLL drivers** (included in the companion CD-ROM)
- **Wiring cable** PCL-10168
- **Wiring board** PCLD-8710, ADAM-3968

- **Computer** personal computer or workstation with a PCI-bus slot (running Windows 95/98/NT/2000/XP)

Some other optional components are also available for enhanced operation

- Application software ActiveDAQ, GeniDAQ or other third-party software packages.

After getting the necessary components and maybe some of the accessories for the enhanced operation of the Multifunction card, begin the Installation procedures. Figure 3.2 provides a concise flow chart to give a broad picture of the software and hardware installation procedures:

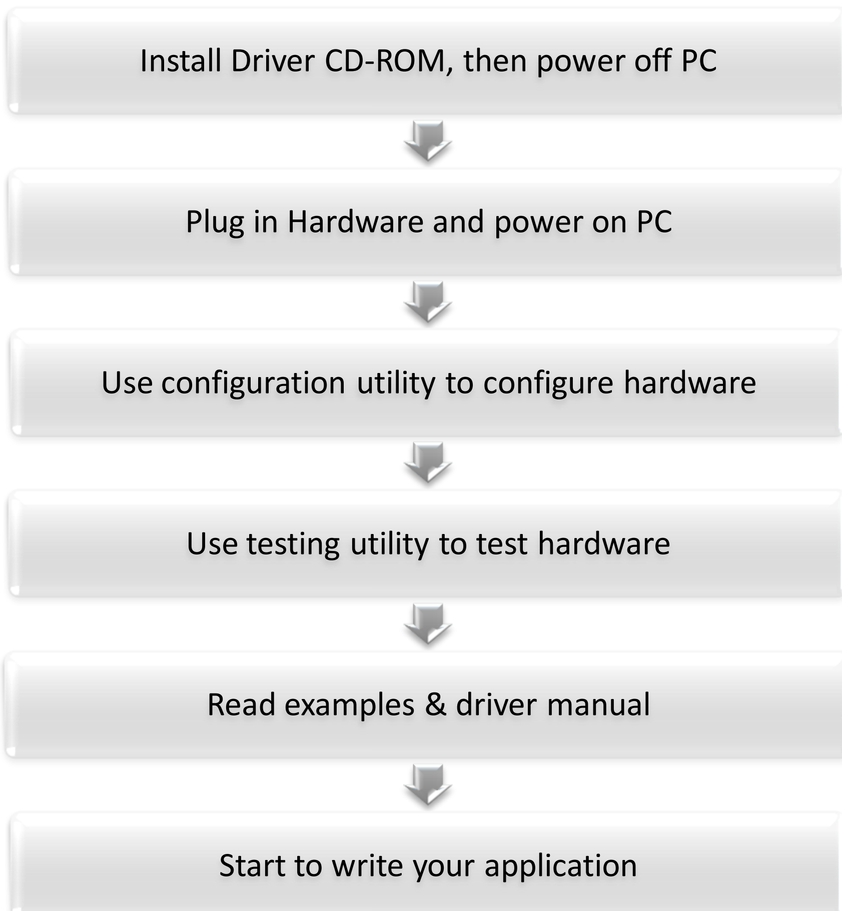


Figure 3.2: PCI-1710HG Installation Flow Chart

Advantech offers a rich se of DLL drivers, third party driver support and application software to full exploit the functions of the PCI-1710HG card:

- DLL driver (on the companion CD-ROM)
- LabVIEW driver
- Advantech ActiveDAQ
- Advantech GeniDAQ

Real time computing is the study of hardware and the software systems that are subject to a “real-time constraint” example, operational deadline from event to system response. A *non real-time system* is one for which here is no deadline, even if a fast response or high performance is desired or even preferred. The needs of real-time software are often addressed in the context of real time operating system, and synchronous programming languages, which provided guide on which to build real time application software

.

A real-time system may be one where its application can be considered to be mission critical. The anti-lock brakes on a car are an example of a real-time computing system; the real-time constraint in this system is the short time in which the brakes must be released to prevent the wheel from locking. Real-time computations can be said to have failed if they are not completed before their deadlines, where their deadlines is relative to an event. A real-time deadline must be met, regardless of system load.

The term real-time derives from its use in early simulation. While current usage implies that a computation that is ‘fast enough’ is real-time, originally it referred to a simulation that proceeded at a rate that of the real process it was stimulation. Analog - computers, especially, were often capable of stimulating much faster than real-time, a situation that could be just as dangerous as a slow simulation if it were not also recognized and accounted for.

Real time computing is sometimes misunderstood to be high-performance computing, but this is not always the case. For example, a massive supercomputer executing a scientific simulation may offer impressive performance, yet it is not executing a real-time computation. Conversely, once the hardware and the software for an anti-locking braking system has been designed to meet its required deadlines, no further performance gains are necessary. Furthermore, if a network server is highly loaded with network traffic, its response time maybe slower but will still be succeeded.

Hence, such a network server would not be considered a real-time system, temporal failures (delays, time-outs, etc.) are typically small and compartmentalized (limited in effect) but are not catastrophic failure. In a real-time system, a slow-down beyond limits would often be considered catastrophic in its application context. Therefore, the most important requirement of a real-time system is predictability and not performance.

Some kind of software, such as many chess-playing programs, can fall into either category. For instance, a chess program designed to play in a tournament with a clock will need to decide on a move before a certain deadline or lose the game, and is therefore a real-time computation. But a chess program that is allowed to run indefinitely before moving is not. In both of these cases, however, high performance is desirable, the more work on a tournament chess program can do in the allotted time, the better its moves will be, and the faster an unconstraint chess program runs, the sooner it will be able to move. This example also illustrates the essential difference between real-time computations and other computations. If the tournament chess program does not make a decision about its next move in its allotted time, it loses the game, example if it fails as a real-time computation – while in the other scenario, meeting the deadline is assumed not to be necessary.

3.3.3 Robotic arm with direct current (DC) geared motor

The robotic arm has been built specially for this project. The procedure of building this robotic arm will be shown below (figure 3.2 to 3.9) the details of this robotic arm are also attached below.

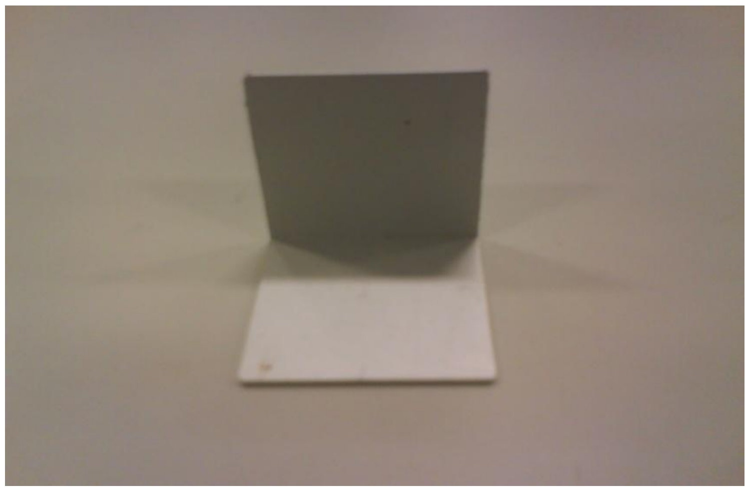


Figure 3.3: the case for motor.

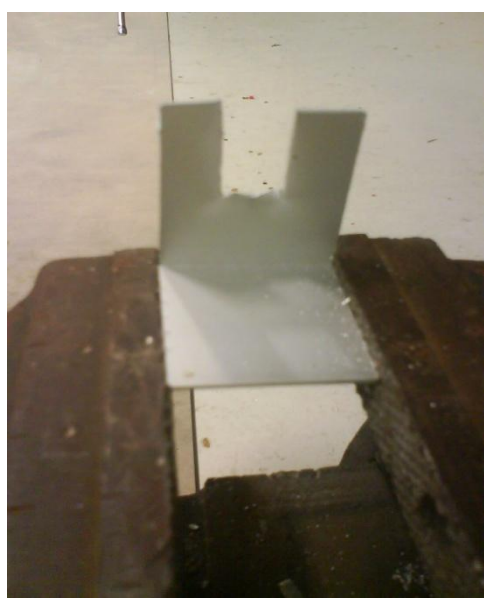


Figure 3.4: the case has been holed so that the motor could fit in.

Figure 3.5: the steel is being measured

Figure 3.6: the steel have been cut

Figure 3.7: both the steel have been welded together to make the robotic arm base

Figure 3.8: the finished robotic arm based

Figure 3.9: the arm for the robotic arm. Webcam will be attached on this arm.

Figure 3.10**:** the motor have been attached together with the base and the robotic arm. However the webcam still is not attached.

3.3.4 Webcam

The standard webcam is being used in this project. The webcam have 30 frames per second (fps) and work excellently for this study. The webcam that is being used in this study is Creative Live! Cam Chat USB. No driver installations are needed thus it is hassle-free to use. It has a built in microphone for added convenience and the video resolution up to 800 x 600 pixels.

The webcam will be mounted on the robotic arm (figure 3.8). The webcam's USB will be attached to the PC. Using image processing, it will read the environment. If there is a movements, than the camera will captured it and the system will process it. After the system has process the information, it will give feedback to the robotic arm on whether it needed to stay or turn to left or right according to the movements.

Figure 3.11: the Creative Live! Cam Chat USB

3.4 SOFTWARE ANALYSIS

The software that is being used is image processing in MATLAB and also data acquisition card (DAQ) software.

3.4.1 Image processing in MATLAB

Image processing toolbox could be used for image enhancement, reducing the noises and motion, design and implement 2D spatial and frequency filters, to isolate region of interest, identify image of objects, and to track motion of an object in series of images.

In this study, detecting and recognizing is essential so that any suspicious movements could be tracked. Image subtraction is used for detecting changes.

$$g(x, y) = f(x, y) - h(x, y)$$

$f(x, y)$ = background image or stationary image
$h(x, y)$ = frames after background image
$g(x, y)$ = changes in environment, resulting in suspicious movements.

The process on how the image processing works is shown below (figure 3.12)

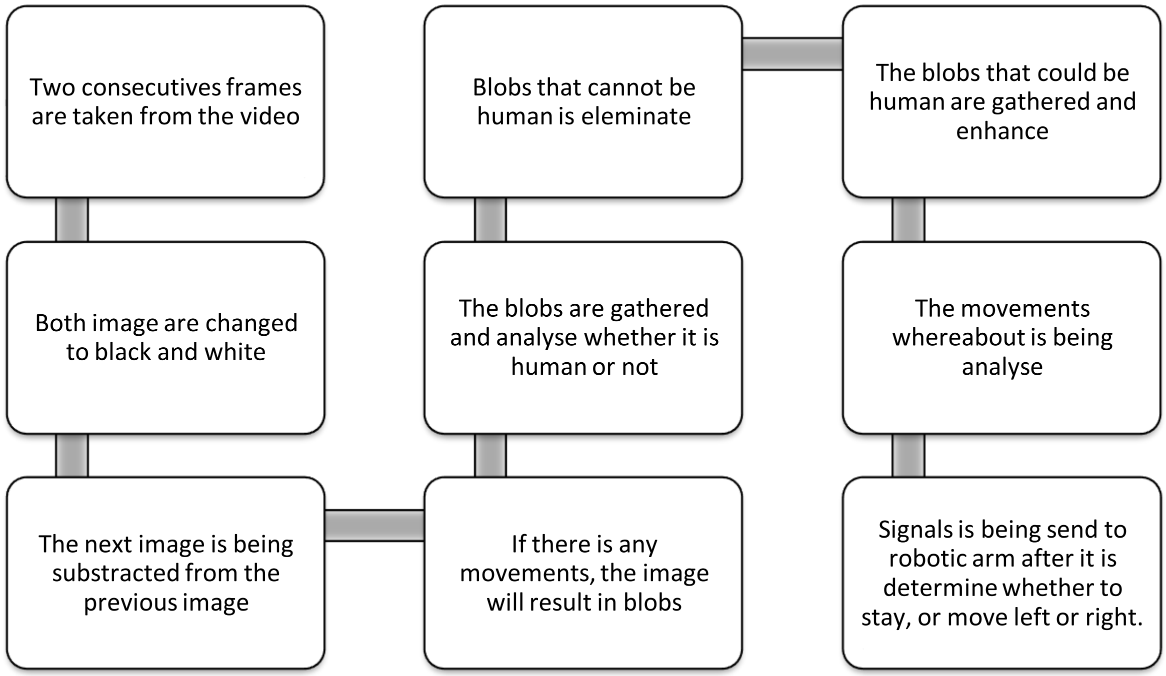


Figure 3.12: Process flow of image processing

3.4.1.1 Image Processing Technique

Although there are many techniques on image processing, the one used in this system is background subtraction algorithm or BSA. First, a video of a controlled room is recorded (figure 3.13). The video is then being decompressed and it is converted to a set of pictures. The first chosen frame is frame 32 where the room is isolated, and have many objects such as bag, chairs, switch and so on. A person then entered the room in frame 37 (figure 3.14). The system then subtracts the image (after) with the image (before) (figure 3.15). This type of image is then converted to a binary image so that it is easier to count the pixel or blobs (figure 3.16). The blobs are then counted as a sum. After running 40 images, it could be concluded that a suspicious movements such as a human figure has a sum of more than 1000.

Figure 3.13: a controlled room (frame 32)

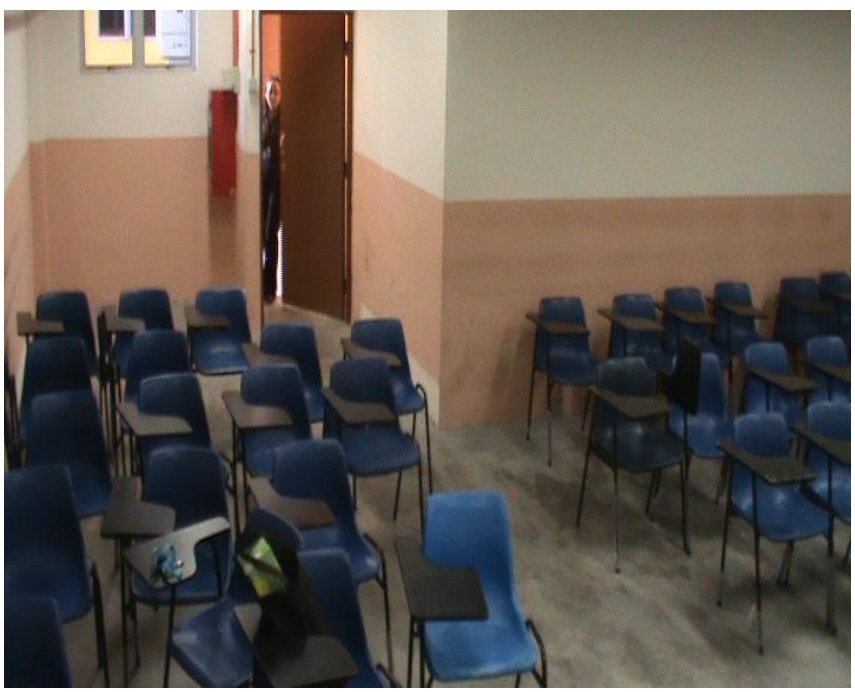


Figure 3.14: a person entered the isolated room (frame 37)

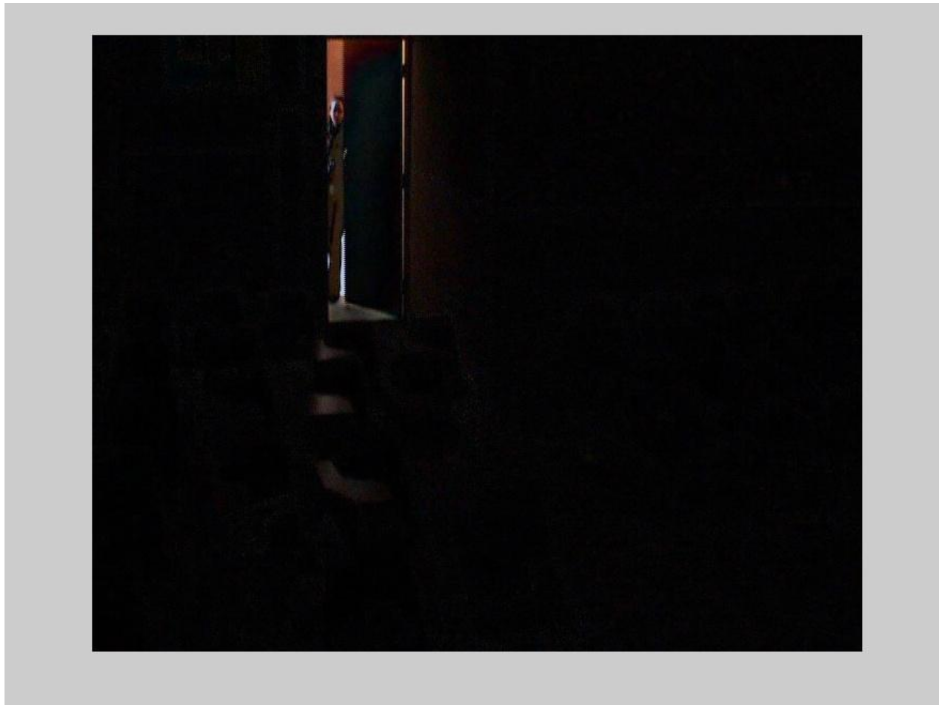


Figure 3.15: the subtracted image of frame 32 and 37.

Figure 3.16: the image has been converted to a binary image.

3.4.2. Data Acquisition Card (DAQ) software

This software is a must so that the data acquisition card could be used. Without this driver, the data acquisition card could not be used.

3.4.3 Real Time Window Target

Real-time Window Target™ rapid prototyping software is a PC solution for prototyping and testing real-time systems. Real-Time Windows Target® software uses a single computer as a host and target. On this computer, MATLAB environment, Simulink® software and Stateflow® software (optional) is use to create models using Simulink blocks and Stateflow diagrams.

After creating a model and simulating it using Simulink software in normal mode, it can generate executable code with Real-Time Workshop code generation software, Stateflow Coder code generation software (optional), and the Open Watcom C/C++ compiler. Then the application can be run in real time with Simulink external mode.

Real-Time Windows Target uses standard and inexpensive I/O boards for PC-compatible computers. When running the models in real time, Real-Time Windows Target captures the sampled data from one or more input channels, uses the data as inputs to the block diagram model, immediately processed the data, and sends it back to the outside world through an output channel on the I/O board.

Real-Time Windows Target provides a custom Simulink block library. The I/O driver block library contains universal drivers for supported I/O boards. These universal blocks are configured to operate with the library of supported drivers. This allows easy location of driver blocks and easy configuration of I/O boards.

It only needs to drag and drop a universal I/O driver block from the I/O library the same way as it would from a standard Simulink block library. And it connects an I/O driver block to the model just as it would connect any standard Simulink block.

It just needs to create a real-time application in the same way as it creates any other Simulink model, by using standard blocks and C-code S-functions. It can add input and output devices to the Simulink model by using the I/O driver blocks from the rtwinlib library provided with Real-Time Windows Target. This library contains the following blocks:

- Analog input
- Analog output
- Counter input
- Digital input
- Digital output
- Encoder input
- Frequency output
- Packet input
- Packet output
- Stream input
- Stream output

3.4.3.1 Setup and Configuration

Real-time Window Target can use any PC compatible computer that runs Windows 2000, Windows XP 32-bit or Windows Vista 32-bit. The computer can be a desktop, a laptop or notebook PC.

3.4.3.1.1 Compiler

Compiler code is created from the generated C-code using the Open Watcom C/C++ compiler. For convenience, this compiler is shipped with the Real-Time Windows Target software. No other third-party compiler is needed or can be used.

The Real-Time Windows Target software always uses the Open Watcom C/C++ compiler, even if you have specified some other compiler using the mex-setup command. Real-Time Windows Target software cannot be configured to use a compiler other than Open Watcom C/C++.

3.4.3.2 Kernel Setup

During software installation, all Real-Time Windows Target software is copied onto the hard drive, but the Real-Time Windows Target kernel is not automatically installed into the operating system. The kernel must be installed before a Real-Time Windows Target application can be run. Installing the kernel configures it to start running in the background each time the computer is start. The following procedure describes how to use the command `rtwintgt –intall`. The command `rtwintgt –setup` can also be used instead. To install the kernel:

1. `Rtwintgt –install`

   is typed in the MATLAB Command Window

   Or:

   a. Click the MATLAB **Start** button

   b. Select **Link and Targets > Real-Time Windows Target > Install real-time kernel**

The MATLAB Command Window display one of these messages:

```
You are going to install the Real-Time Windows Target
kernel.
Do you want to proceed? [y]
```

Or:

```
There is a different version of the Real-Time Windows
Target kernel installed.
Do you want to update to the current version? [y] :
```

2. `y` is typed to continue installing the kernel, or **`n`** to cancel installation without making any changes.

   If y, the MATLAB environment installs the kernel and displays the message:

   ```
   The Real –Time Windows Target kernel has been
   successfully installed.
   ```

3. If a message appeared asking to restart your computer, do so before attempting to use the kernel, or the Real-Time Windows Target model will not run smoothly.

4. After installing the kernel, verify that it was correctly installed by typing:

   ```
   rtwho
   ```

   The MATLAB Command Window should display a message that shows the kernel version number, followed by performance, timeslice and other information similar to below:

```
Real-Time Windows Target version 3.5.0 (C) The
MathWorks, Inc. 1994-2010
Running on Multiprocessor APIC computer.
MATLAB performance = 100.0%
Kernel timeslice period = 1ms
```

Once the kernel is installed, you can leave it installed. The kernel remains idle after you have installed it, which allows the window operating system to control the execution of any standard Windows based application, including Internet browsers, word processors, the MATLAB environment, and so on. The kernel becomes active when the user begins to execute of their model, and becomes idle again after model execution completes.

3.4.3.3 Testing the Installation

Once the installation of the Real-Time Windows Target software and kernel is completed, it is recommended a quick test by running the model rtvdp.mdl. Doing this test is a quick check to confirm that the Real-Time Windows Target software is still working. The model rtvdp.mdl does not have any I/O blocks, so that this model can run regardless of the I/O boards in your computer. Running this model will test the installation by running Real-Time Workshop code generation software, Real-Time Windows Target software, and the Real-Time Windows Target kernel.

1. To open the demo model `rtvdp` is typed in the MATLAB Command Window or launch MATLAB Online Help and choose **Demos** > **Links and Targets** > **Real-Time Windows Target** > **Real-Time Van der Pol Simulation.**
   The Simulink model `rtvdp.mdl` window opens (figure 3.17)

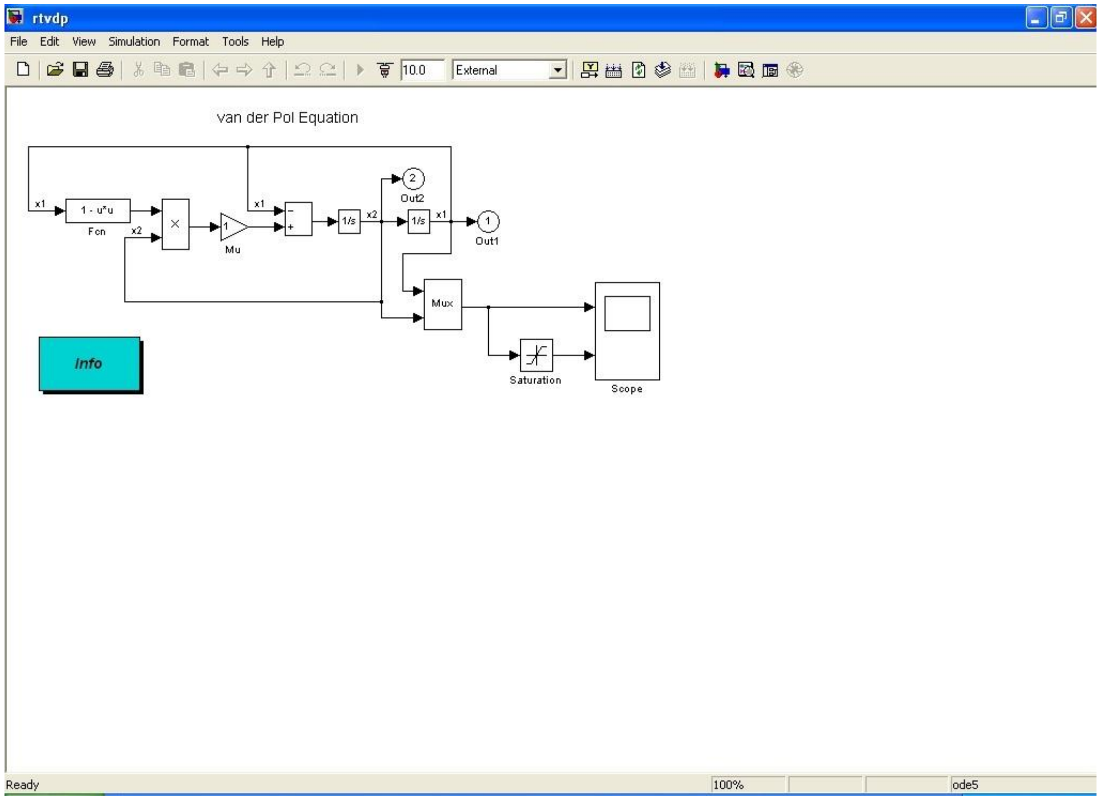


Figure 3.17: Simulink Model rtvdp.mdl

2. From the **Tools** menu, **Real-Time Workshop** is chosen and **Build Model** is selected. The MATLAB Command Window displays the following messages:

```
###  Starting Real-Time Workshop build for model
:rtvdp
### Invoking Target Language Compiler on rtvdp.rtw
. . .
### Compiling rtvdp.c
. . .
### Created Real-Time Windows Target module
rtvdp.rwd.
### Successful completion of Real-Time Workshop
builds procedure
For model :rtvdp
```

3. From the **Simulation** menu, click **External**, and then click **Connect to target**.
   The MATLAB Command Window displays the following message:

   ```
   Model rtvdp loaded
   ```

4. From **Simulation** menu, **Start Real-Time Code** is clicked.
   The Scope window displays the output signals. If the Scope window looks like the next figure, the Real-Time Windows Target software has successfully installed and has run a real-time application (figure 3.18).

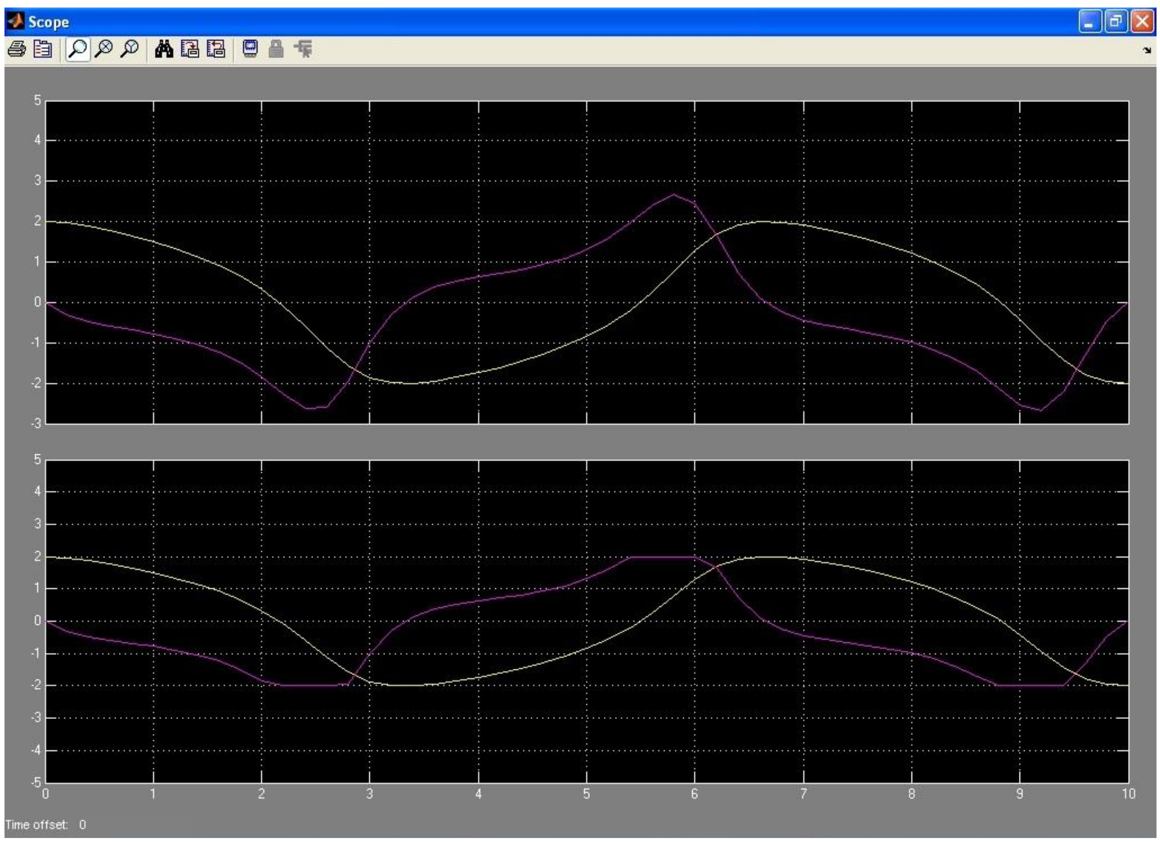


Figure 3.18: Output Signal rtvdp.mdl

5. From Simulation menu, Stop Real-Time Code is selected. The real-time application stops running and the Scope window stops displaying the output signals.

3.4.4 Creating a Real Time Application

This procedure explains how to create a simple Simulink model. This model is used as an example to learn other procedures that are useful with Real-Time Windows Target software. A Simulink model is created before running a simulation or creates a Real-Time Target software

1. In the MATLAB Command Window, `simulink` is typed
   The Simulink Library Browser opens (figure 3.19). The left pane shows a hierarchy of libraries and block categories, with the Simulink library at the top. The right pane shows the blocks available in the category selected on the left. See “Library Browser” for more information.

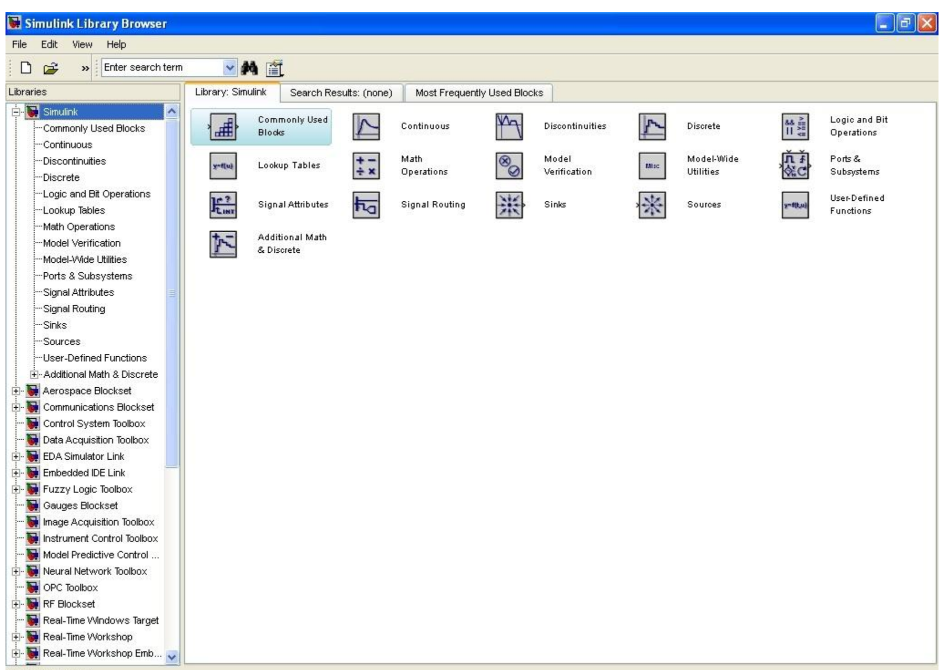


Figure 3.19: Simulink Library Browser

2. Choose **File > New > Model**, or the **New model** button is click on the toolbar. An empty Simulink window opens (3.20):

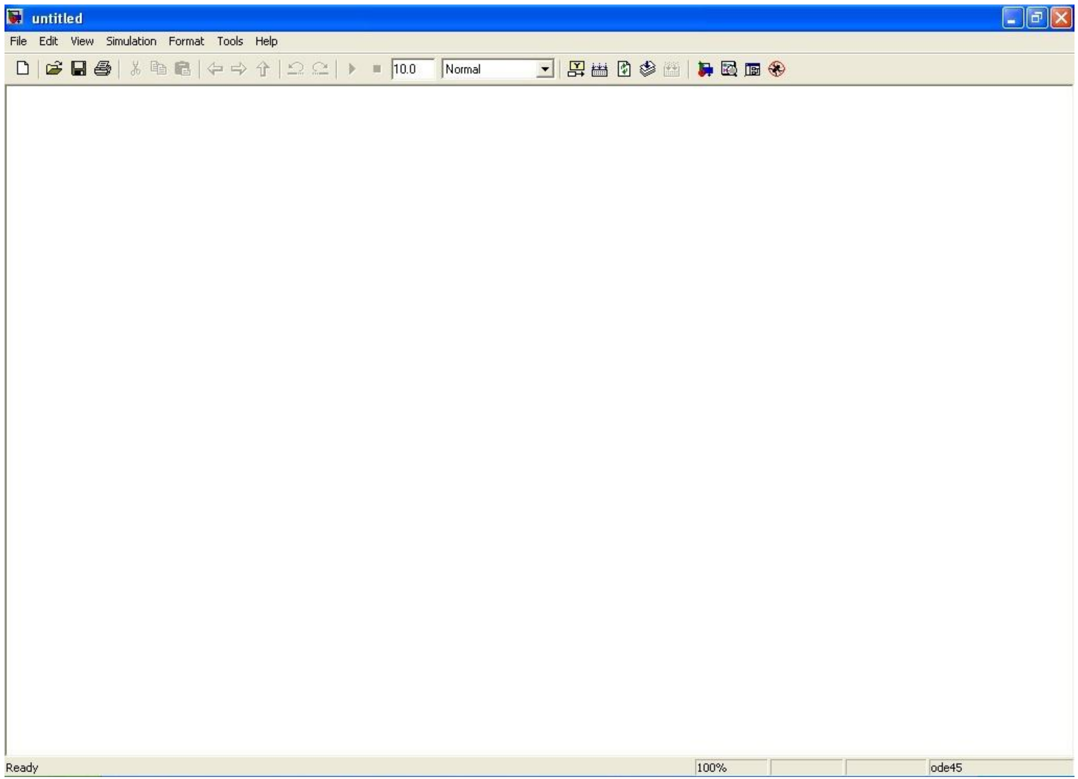


Figure 3.20: Empty Simulink Windows

3. In the left pane of the Simulink Library window, **Simulink** is click and **Sources** is chosen. Then **Signal Generator** blocks is click and drag from the browser to the Simulink window. From **Sinks, Scope** is click and drag to the Simulink window. Real-Time Window is selected and **Analog Output** is drag to the Simulink window.

4. The output of the **Signal Generator** is then connected to the **Scope** input by clicking and dragging a line between the blocks. Likewise, the **Analog Output** is connected between the **Scope** and **Signal Generator**.

5. Signal Generator block is double clicked. The Block Parameters dialog box opens. From the **Wave form** list, square is chosen.
   In the Amplitude text box, 1 is entered.
   In the Frequency text box, 20 is entered.
   From the Units list, rad/sec is selected.
   The Block Parameters dialog box shown in figure below (3.21):

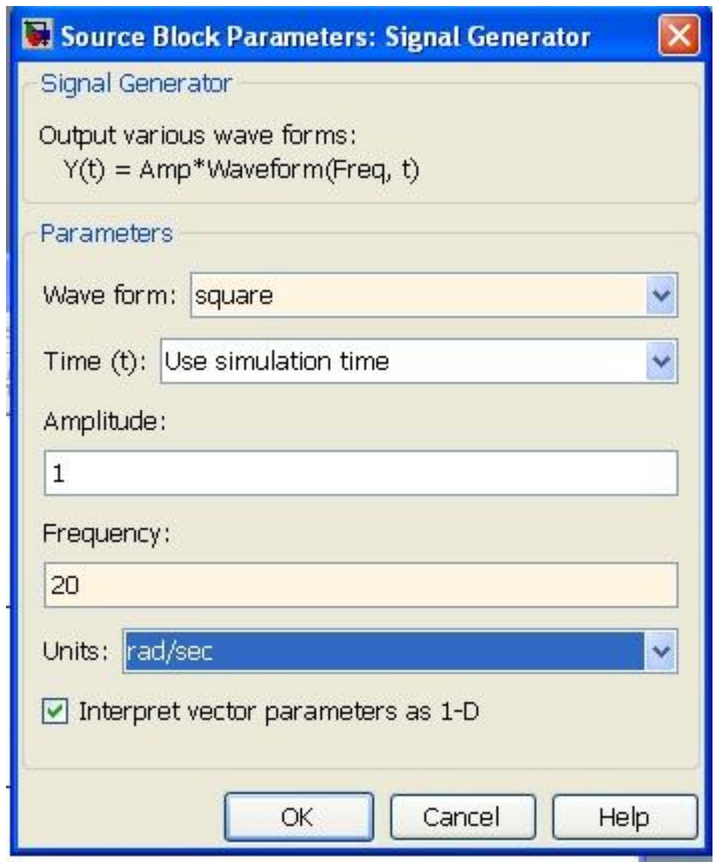


Figure 3.21: Signal Generator Block Parameter

6. **OK** is clicked.

7. The **Analog Output** block is double clicked.

The Analog Output Block Parameters dialog box opens. The I/O Block parameters dialog box opens. For an Analog Output block, the dialog box is shown in figure 3.22

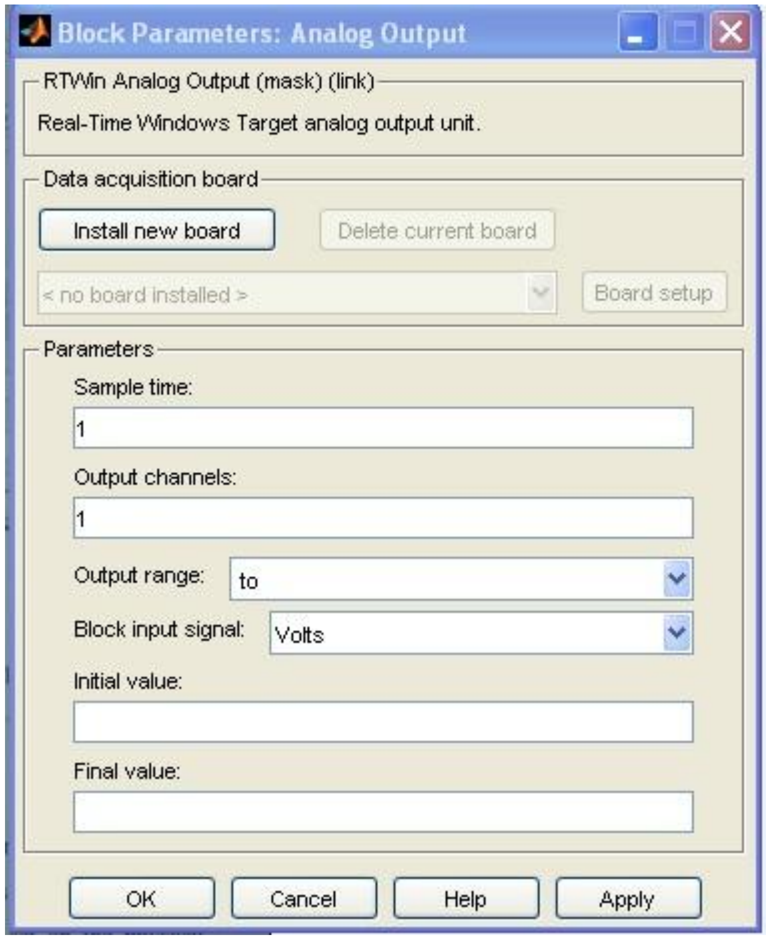


Figure 3.22: Analog Output Block Parameters

8. **Install new board** is clicked. From the list that appears, the manufacturer of the board is pointed and then a board type is selected. For example, it pointed to **Advantech**, then click **PCI-1710HG**.

9. One of the following is selected, as appropriate to the board:

    - For an ISA bus board, a hexadecimal base address is entered. This value must match the base address jumpers or switches set on the physical board. For example, if a base address of `0x300` is entered, in the **Address** box `300` is typed. The base address is selected by checking boxes **A9** through **A3**
    - For a PCI bus board, the **PCI slot** is entered or **Auto-detect** are checked. (figure 3.23)

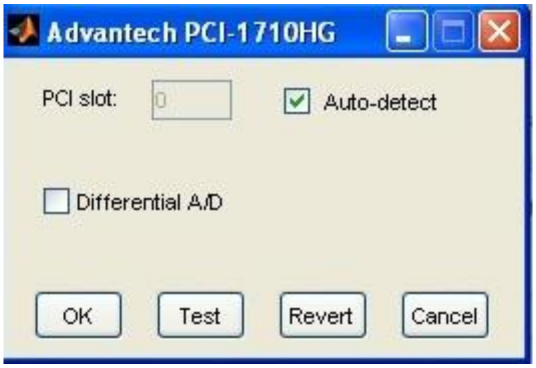


Figure 3.23: Advantech PCI-1710HG

10. The Block Parameters dialog also able to set other block parameters, such as the sample time. Set such parameters as needed.

11. The **Test** is clicked.

    The Real-Time Windows Target kernel tries to connect to the selected board, and if successful, displays the following message:

Figure 3.24: Board Test OK Dialog

12. **OK** button on the message box is clicked, and again on the Block Parameters dialog box.

    The I/O Block Parameters dialog box closes and the parameter values are included in the Simulink model.

13. In the **Sample time** box, enter the same value entered in the **Fixed step** size box from the Configuration Parameters dialog box, or an integer multiple of that value `0.001`.

14. In the **Output channels** box, a channel vector is entered that selects the analog output channels used on this board. The vector can be any valid MATLAB vector form. For example, to select single analog output channels on the **PCI-1710HG** board, `1` is entered or to select both analog output channels, `[1,2]` or `[1:2]` is entered.

15. For the Output range list, the input range for all of the analog input channels entered in the Input channels box is copied. For example, with the **PCI-1710HG** board, 0 to 10V is chosen.

16. From the Block input signal list, choose from the following options:

    - Volts – Expects a value equal to the analog voltage.
    - Normalized unipolar – Expects a value between 0 and +1 that is converted to the full range of the output voltage regardless of the output

voltage range. For example, an analog output range of 0 to +5 volts and -5 volts to +5 volts would both be converted from values between 0 to +1.

- Normalized bipolar – Expects a value between -1 and +1 that is converted to the full range of the output voltage regardless of the output voltage range.
- Raw – Expects a value of 0 to 2n – 1. For example, a 12-but A/D converter would expect a value between 0 and 2(12) – 1(0 to 4095). The advantage of this method is the expected value is always an integer with no round off errors.

17. The initial value for each analog **Output channel** entered in the Output channels box. For example, if 1 was entered in the **Output channels** box, and an initial value of 0 volts is needed, enter 0.

18. The final value is entered for each analog channel entered in the **Output channels** box. For example, if `1` is entered in the **Output channels** box, and a final value 0 volts is needed, `0` is entered.

19. One of the following is clicked:

    - **Apply** to apply the changes to your model and leave the dialog box open.
    - **OK** to apply the changes to your model and close the dialog box.

20. In the Simulink window, the Scope block is double click.
    A Scope window opens.

21. The Parameter button is click.
    A Scope parameters dialog box is opens.

22. The **General tab** is clicked. In the **Number of axes** field, the number of graphs needed in one Scope window is entered. For example, 1 is entered for a single

graph. Do not select the floating scope check box. In the **Time range** field, the upper value for the time range is entered. For example, 1 second is entered. From the **Tick labels** list, bottom axis only is chosen. From the **Sampling** list, **decimation** is chose and number 2 is entered in the text box. The Scope parameters dialog box looks similar to Figure 3.25.

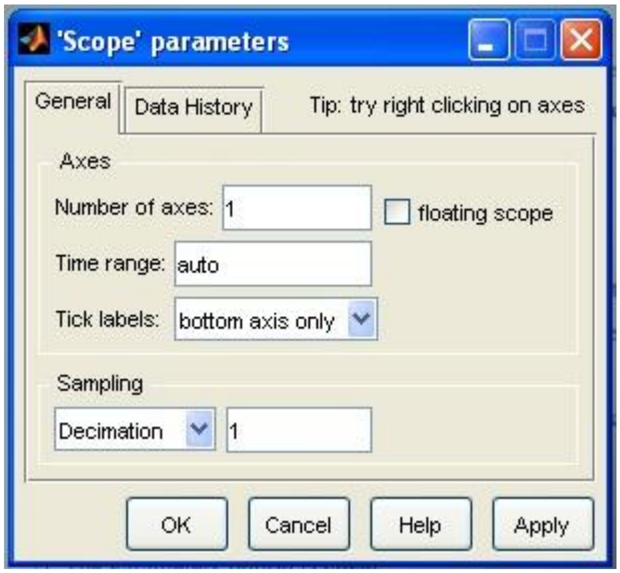


Figure 3.25: Scope Parameters Dialog Box

23. One of the following is clicked:
    - **Apply** to apply the changes to your model and leave the dialog box open.
    - **OK** to apply the changes to your model and close the dialog box

24. In the Scope window, the y-axis is right clicked. From the menu, **Axes Properties** is clicked.

    The Scope properties: axis 1 dialog box opens.

25. In the **Y-min** and **Y-max** text boxes the range for the y-axis is entered in the Scope window. For example, −2 and 2 is entered.

26. One of the following is clicked:

- **Apply** to apply the changes to your model and leave the dialog box open.
- **OK** to apply the changes to your model and close the dialog box

The complete Simulink block diagram is shown in Figure 3.26:

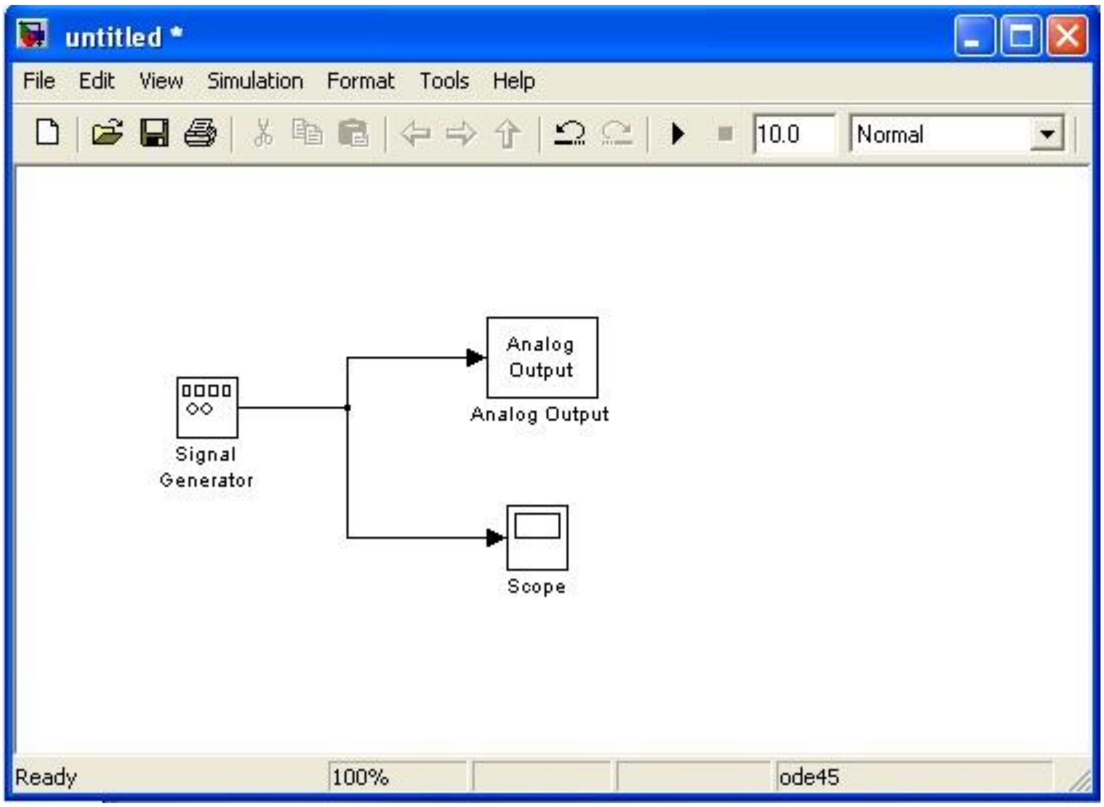


Figure 3.26**:** Completed Simulink Block Diagram

27. From the **File** menu, Save as is clicked. The Save As dialog box open. In the **File name** text box, a filename for the Simulink model is entered and **Save** is clicked. For example, `rtwin_model` is typed.

The Simulink software saves the model in the file rtwin_model.mdl.

CHAPTER FOUR

RESULT AND DISCUSSION

4.1 Introduction

This chapter present about the results that were gain during the development of the system with details and explanation, and also the discussion on this system.

4.2 Results

The system is developed so that it could detect suspicious movement in its surveillance. Therefore, it is vital that it could decide whether it could detect any suspicious movement or not.

A video of a person walking around in a controlled environment is recorded and feed into the system. The system starts to calculate the difference between frames, and start showing the results immediately. The suspicious movement will be decided as the silhouette will show what it actually is. The environment is divided into two portions, the left side and the right side. If the suspicious movement is in the left side, the system will alert "intruder on the left side!" and if the suspicious movement is on the right side, the system will alert "intruder on the right side!" If there is no suspicious movements, or the sum of the pixel is small enough or negligible, it will show as no object found. Figure 4.1 below shows the video sequence of one person walking around in the controlled room.

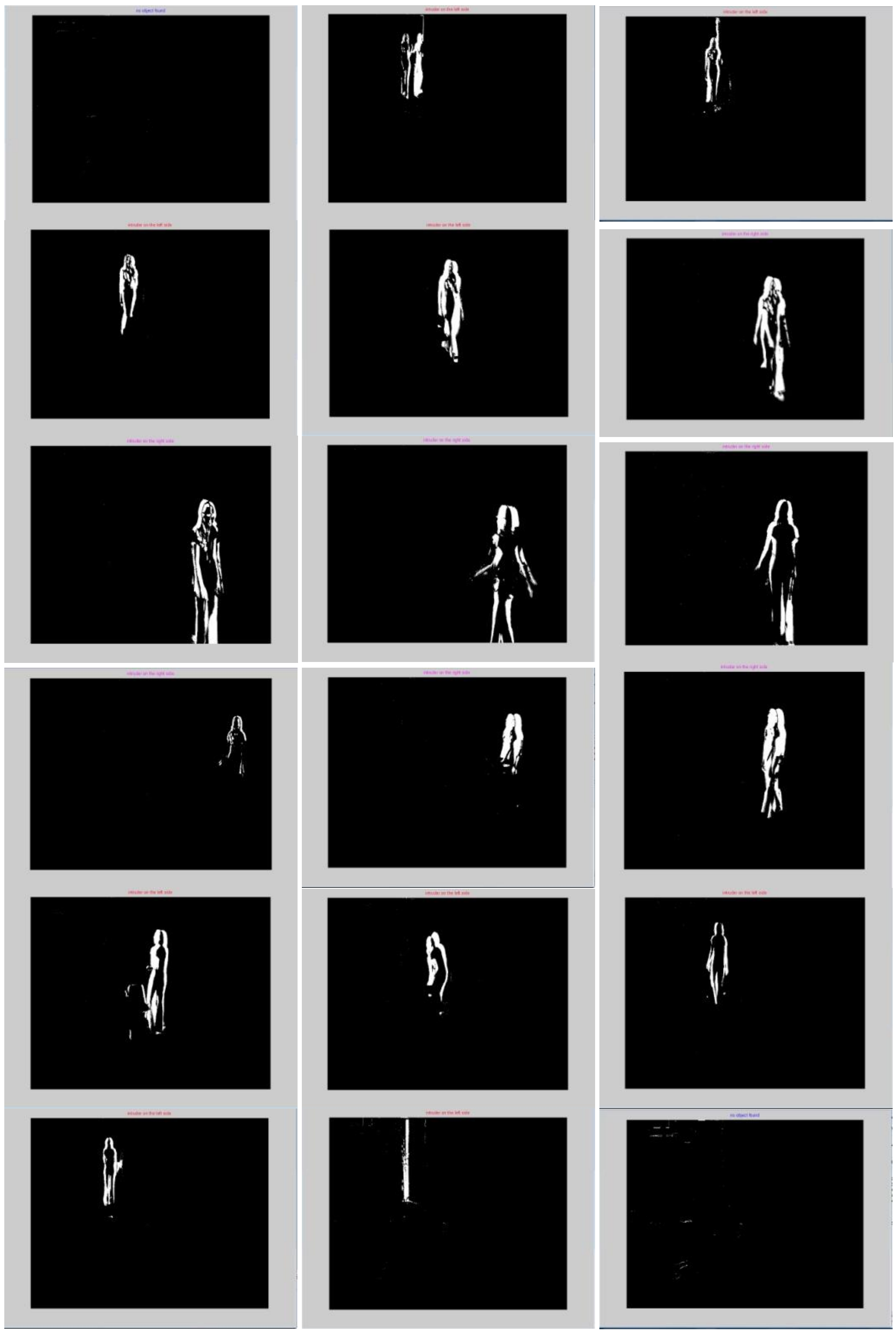


Figure 4.1: a video sequence of a person walking around in a controlled environment.

Figure 4.1 shows that a video sequence of an intruder walking around in the room. The intruder starts entering from frame 37 and the system starts subtracting the image from the last frame before it. Thus from here, it could be concluded that the intruder enters from the door as we could see the door silhouette. The intruder then start walking around from left region towards the right region and walks back towards the door and exited. The images are taken from 200 frames that are converted from the video.

Figure 4.2: Figure above shows a controlled environment, where no suspicious movement detected, or negligible. The title says “no object found”

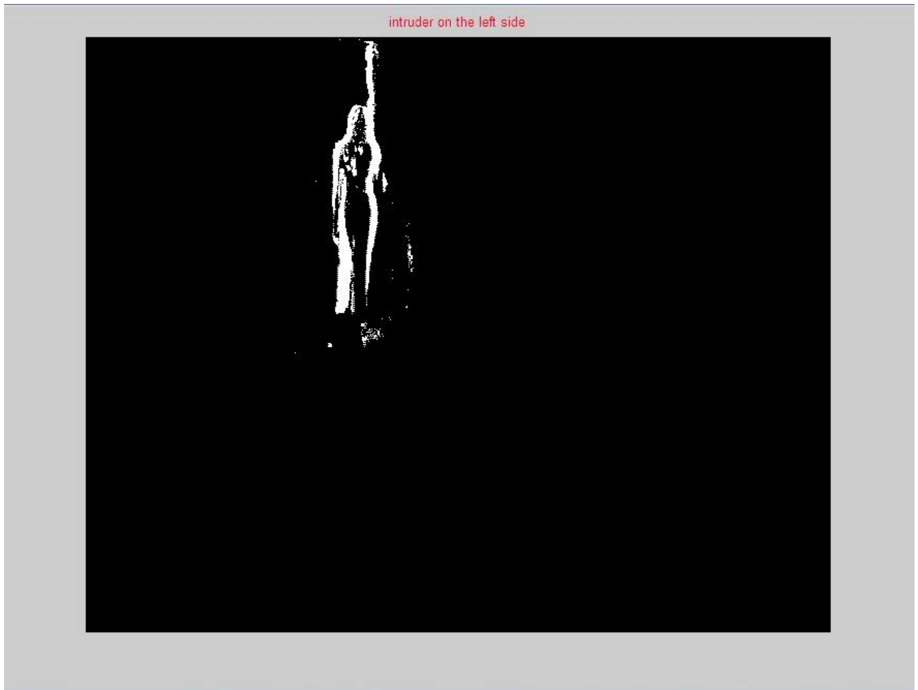


Figure 4.3: Figure above shows when a suspicious movement entered the controlled environment, n thus a warning “intruder on the left side” appeared.

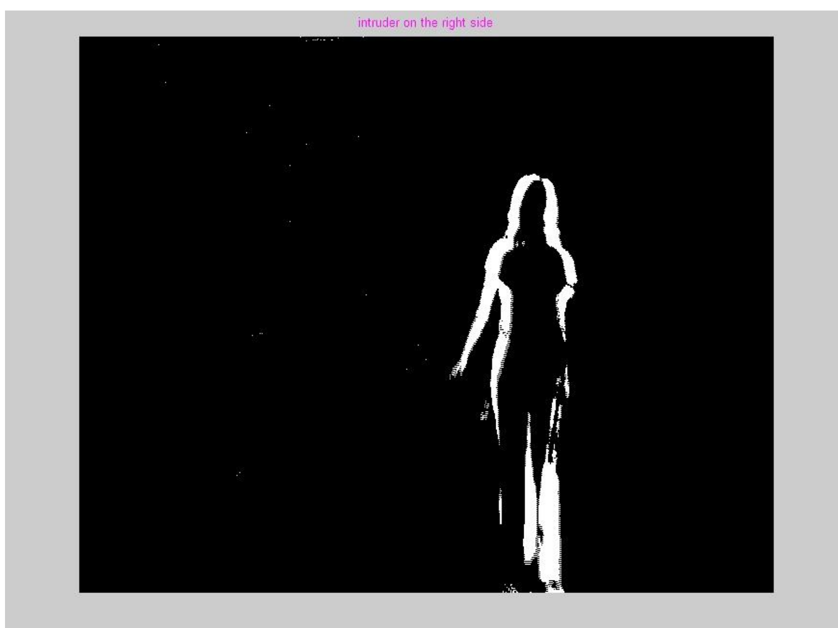


Figure 4.4**:** Figure above shows when an intruder is on the right side. The warning “intruder on the right side” appeared on top of the video.

Figure 4.5: Blobs which considered as intruder as the intruder try to open the door to the isolated room.

Figure 4.6: blobs which are not considered as intruder and thus, negligible.

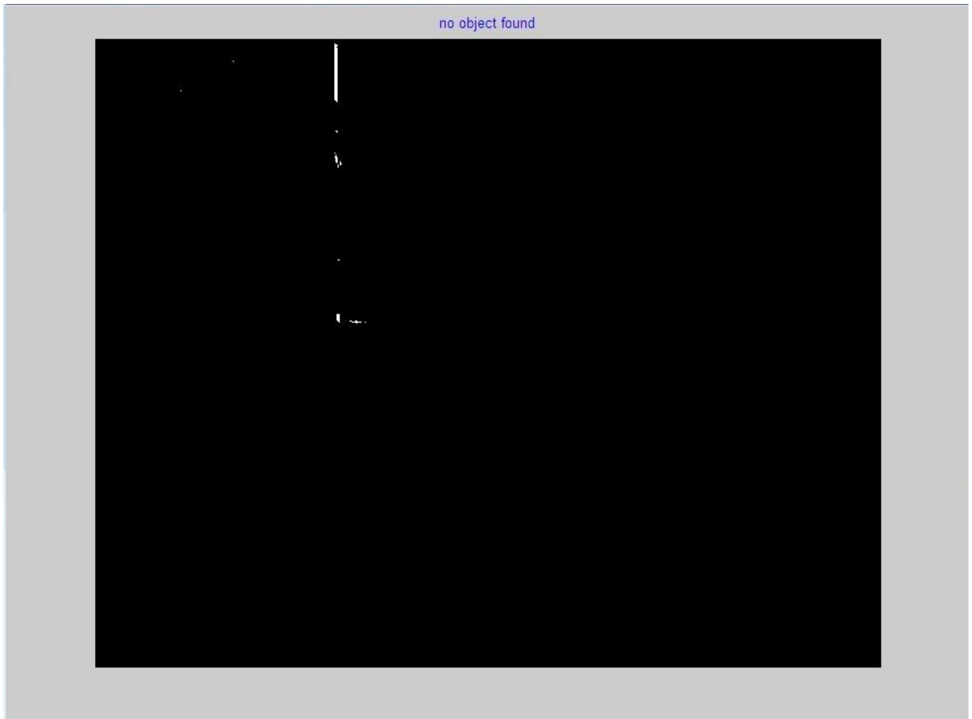


Figure 4.7: blobs which is not considered as intruder.

4.3 Effectiveness and Accuracy

$$accuracy(\%) = \frac{numberofimage\ corrcetly\ identified}{totalnumber\ of\ image}\ x\ 100$$

$$\frac{200}{200}\ x\ 100 = 100\%$$

$$error(\%) = \frac{number\ of\ wrongly\ identified\ image}{total\ number\ of\ image}\ x\ 100\%$$

$$\frac{0}{200}\ x\ 100\% = 0\%$$

At the end of the process, the value of accuracy could finally be calculated. By analysis and calculation, it shows that the accuracy value is 100% of correctness with 0% of error. When the system read an image it could correctly detect intruder and label it whether it is in left or right region. From the analysis, it is clear that the objective is achieved as a system with a high accuracy of detecting and tracking is realized.

4.4 Discussion

Detecting and tracking system is vital for security measurement and thus could be developed more. Every man's meat is other man's poison, and for that, this system has its own advantage and disadvantage.

4.4.1 Advantages

Object detecting tracking is vital for security measurement as follow:

1. Image processing is perfect for detecting and tracking as the coding could easily manipulated to our advantage.
2. Background subtracting algorithm could easily detect suspicious movements that is out of order. Missing things could also be detected by using background subtracting.
3. The need of using officer could be reduce or even eliminated so it is cost savvy.
4. The system is could be alert 24hours unlike human operators that tend to fall asleep or tired.
5. The system could further improve and improvise with the technologies that are available or merge with other technologies.

4.4.2 Disadvantages

1. The camera sometimes does move, so the output image could detect small blobs that actually belong to the environment.
2. Other object such as large animals could be perceived as intruder too
3. Low resolution camera makes the detection hard to do.

This system is very useful to the world as its current application as security measurement. These include premises such as banks, airport, and small shops and even in houses. All this application implements a high security and safety factor.

All the process that has been passing through the system had given a good result with high accuracy and it is supported by theoretical facts and knowledge and also analytical analysis. The image acquisition is successfully loaded to the system and it could be accepted by the system. It is then proceed with process of pre-processing whereby there are several steps to be considered such as image subtraction, thresholding, and binary conversion. All of these processes have special command for each and every of them. The preprocessing is a very important method in order to proceed with future extraction process. The output results have given appropriate command and labeled for each object detected. The method that is implemented has successfully obtained the required result.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1 Introduction

This system is fully functional to detect and tracking suspicious movements. The system also did achieve its objective on detecting and tracking suspicious movement. This chapter summarize on overall of this system, and also recommendation and suggestion to further improve this system.

5.2 Conclusion

Suspicious movements on a premise are usually unwanted, and thus, this system could be fully implemented on such premises. Based on the analysis that has been done, this system could detect 100% suspicious object on a controlled environment. The system is also eliminates blobs that is formed because the objects in the environment such as chairs and so on. It tracks the movements and labeled it as in the right or left region and commands the motor of the robot to move according to the region where the movements are.

Based on above, the conclusions for this system are:

1. The system that is being built is reliable for detecting suspicious movements with a high reliable percentage.
2. The suitable value for a certain process will determine the effectiveness of this system. For example a sum of a value under 600 could detect environment, and not only objects, and also small objects such as animal. But a sum value over 700 could only detect a big movement, such as human.
3. Image processing is a suitable method in developing this detection system.
4. This detection system can be compute within a few second that shows a high productivity of this system.
5. All the method and analysis in this project had successfully done, therefore it is reliable for any input image.

5.3 Recommendation

For this system we still can make some further study for improvement and upgrading the effectiveness and application. This section describes some idea for extending to upgrade this recognition system in this study. There are:

1. Since a handheld video camera is used to capture the video, the project could be improved by directly capturing the video by using a webcam in the real time. It would be even better if the webcam used is with a high resolution so the image would be crystal clear.
2. Instead of using MATLAB, the system also could be done with other software such as C++ or VisualBasic as both could be interfaced with the hardware. Another suggestion is that the MATLAB code is converted to a C language for a faster and more accurate result.

3. The system could be developed until it could detect and differentiate more than one object and track one of the object and detect what object is missing by differentiate the frames before and after the object is missing.
4. Another improvement that we can add our recognition technique by train data in the neural network system. Know that neural network is a very powerful method whereby the recognition is based on the knowledge that we had gave to the system. Because of that the system will automatically learn about the situation and decision.
5. This is system could be done by increasing the number of images used.

REFERENCES

[1] Citing found from url:

www.NationMaster.com

[2] Citing found from url:

http://en.wikipedia.org/wiki/Closed-circuit_television

[3] Citing found from url:

http://en.wikipedia.org/wiki/11-september

[4] Excerpts taken from www.sciencedirect.com.ezproxy.ump.edu.my

[5] A. Hunter, J. Owens, M. Carpenter, *A neural system for automated CCTV surveillance in S. Velastin (Ed.),* IEE Symposium on Intelligent Distributed Surveillance Systems, February 26, 2003, IEE Savoy Place, London, IEE London.

[6] O. Javed, M. Shah, *Tracking and object classification for automated surveillance,* in Seventh European Conference on Computer Vision, Copenhagen, Denmark, May 28–31, 2002

[7] P. KaewTraKulPong, R. Bowden, *An improved adaptive background mixture model for real-time tracking with shadow detection,* in: Second European Workshop on Advanced Video Based Surveillance Systems, AVBS01, September, 2001.

[8] D. Koller, J. Weber, J. Malik, *Robust multiple car tracking with occlusion reasoning*, in: The third European conference on Computer vision, vol. 1, Stockholm, Sweden, 1994, pp. 189–196.

[9] C. Wren, A. Azarbayejani, T. Darrell, A. Pentland, Pfinder, *Real-time tracking of the human body*, in: IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, 7, 1997, pp. 780–785.

[10] L.M. Fuentes, S.A. Velastin, *Assessment of Digital Image Processing as a means of Improving personal security in public Transport*, Proceedings of the Second European Workshop on Advanced Videobased Surveillance Systems, (AVBS2001) (2001).

[11] S.S. Intille, J.W. Davis, A.F. Bobick, Real-*Time Closed-World Tracking*, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'97) (1997) 697–703.

[12] F. De la Torre, E. Martinez, M.E. Santamaria, J.A. Moran, *Moving Object Detection and Tracking System: a Real-Time Implementation*, in: Proceedings of the Symposium on Signal and Image Processing GRETSI 97, Grenoble (1997).

[13] I. Haritaoglu, D. Harwood, L.S. Davis, *W4: Real-time surveillance of people and their activities*, in: IEEE Transaction Pattern Analysis and Machine Intelligence 22 (8) (2000) 809–822.

[14] S. Huwer, H. Niemann, *Adaptive Change Detection for Real-Time Surveillance Applications*, Proceedings of the IEEE Workshop on Visual Surveillance, Dublin (2000) 37–43.

[15] N. Rota, M. Thonnat, *Video Sequence Interpretation For Visual Surveillance*, in: Proceedings of the IEEE Workshop on Visual Surveillance, Dublin (2000) 59–68.

[16] S. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, H. Wechsler, *Tracking groups of people*, in: Computer Vision and Image Understanding 80 (2000) 42–56.

[17] J.K. Aggarwal, Q. Cai, *Human motion analysis: a review*, Computer Vision and Image Understanding 73 (3) (1999) 428–440.

[18] D.M. Gavrila, *The visual analysis of human movement: A survey*, Computer Vision and Image Understanding 73 (1999) 82–98.

[19] R. Rosales, S. Claroff, *Improved Tracking of Multiple Humans with Trajectory Prediction And Occlusion Modelling,* Proceedings of the IEEE Conference on Computer Vision and Pattern Recodgnition (1998).

[20] J.I. Agbinya, D. Rees, *Multi-object tracking in video*, Real-Time Imaging 5 (1999) 295–304.

[21] P. Peer, F. Solina, *SecurityAgent: Security and Monitoring Computer Enhanced Video System*, Electrotechnical and Computer Science Conference ERK'OI, IEEE Region 8, Volume B, pp. 119-122, Portoroz, Slovenia, 2001. (Slovenian version))

[22] F. Solina, P. Peer, B. Batagelj, S. Juvan, *15 seconds of fame - an interactive, computer vision based art installation*, accepted to International Conferenceon Control, Automation, Robotics and VisionICARCV'02, Singapore, 2002.

[23] P. Peer, F. Solina, *Automatic Human Face Detection Method,* Computer Vision WinterWorkshop CVWW'99, pp. 122-130, Rastenfeld, Austria, 1999.

[24] Kaiqi Huang, LiangshengWang, Tieniu Tan, Steve Maybank, *A real-time object detecting and tracking system for outdoor night surveillance,* Pattern Recognition 41 (2008) 432 – 444, 23 May 2007

[25] Luis M. Fuentesa and Sergio A. Valestin, *People tracking in surveillance applications*, Image and Vision Computing journal 24 (2006). 7 June 2005.

Appendix I

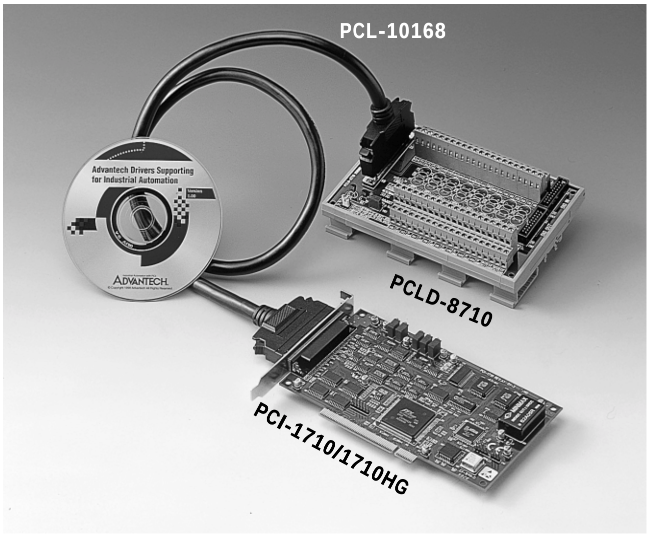
MATLAB program

```
clc
clear all

% how many images needed to be used%

image_num=140;

% referred to the place where the images are

text='D:\7\';

% looping for subtraction and binary conversion

for im=1:image_num
    im1=im+1;
    im;
    filname=strcat(text,int2str(im));
    filname=strcat(filname,'.jpg');
    pic=imread(filname);
    filname1=strcat(text,int2str(im1));
    filname1=strcat(filname1,'.jpg');
    pic1=imread(filname1);
    pic2 = imabsdiff(pic,pic1);
    BW = im2bw(pic2,0.115);
    figure(1),imshow(BW)

  % region left and right
```

```
 c=BW(1:576, 1:359);
 d=BW(1:576, 360:720);

%sum of values of the blobs
 cc=sum(sum(c));
 dd=sum(sum(d));



 hold on

%decision making

 a=(cc>1000)
 b=(dd>1000)

if a>0
    title('intruder on the left side','color','r')

elseif b>0
    title('intruder on the right side','color','m')

else
    title('no object found','color','b')

end

  hold off

end
```

PCI-1710
PCI-1710HG

12-bit, 100 kHz, (High-gain), PCI-bus Multi-function DAS Card



Features

- 16 single-ended or 8 differential analog inputs, or a combination
- 12-bit A/D converter, with up to 100 kHz sampling rate
- Programmable gain for each input channel
- Free combination of single-ended and differential inputs
- On-board 4 K samples FIFO buffer
- Two 12-bit analog output channels
- 16 digital inputs and 16 digital outputs
- Programmable pacer/counter

Introduction

The PCI-1710/1710HG is a multifunction DAS card for the PCI bus. Its advanced circuit design provides higher quality and more functions, including the five most desired measurement and control functions: 12-bit A/D conversion, D/A conversion, digital input, digital output, and counter/timer.

Mixed Single-ended or Differential Analog Inputs

The PCI-1710/1710HG features an automatic channel/gain scanning circuit. The circuit, rather than your software, controls multiplexer switching during sampling. The on-board SRAM stores different gain values and configurations for each channel. This design lets you perform multi-channel high-speed sampling (up to 100 kHz) with different gains for each channel and with free combination of single-ended and differential inputs.

On-board FIFO (First In First Out) Memory

The PCI-1710/1710HG has an on-board FIFO buffer which can store up to 4K A/D samples. The PCI-1710/1710HG generates an interrupt when the FIFO is half full. This feature provides continuous high-speed data transfer and more predictable performance on Windows systems.

On-board Programmable Counter

The PCI-1710/1710HG provides a programmable counter for generating a pacer trigger for the A/D conversion. The counter chip is an 82C54 or equivalent, which includes three 16-bit counters on a 10 MHz clock. One counter is used as an event counter for counting events coming from the input channels. The other two are cascaded together to make a 32-bit timer for a pacer trigger.

Special Shielded Cable for Noise Reduction

The PCL-10168 shielded cable is specially designed for the PCI-1710/1710HG for reducing noise in the analog signal lines. Its wires are all twisted pairs, and the analog lines and digital lines are seperately shielded, providing minimal cross talk between signals and the best protection against EMI/EMC problems.

Specifications

Analog Input:

- **Channels:** 16 single-ended or 8 differential (software programmable)
- **Resolution:** 12-bit
- **On-board FIFO:** 4 K samples
- **Conversion time:** 8 ms
- **Input range:**(V, software programmable)

	PCI-1710	PCI-1710HG
Bipolar	±10, ±5, ±2.5, ±1.25, ±0.625	±10, ±5, ±1, ±0.5, ±0.1, ±0.05 ±0.01, ±0.005
Unipolar	0 ~ 10, 0 ~ 5, 0 ~ 2.5, 0 ~ 1.25	0 ~ 10, 0 ~ 1, 0 ~ 0.1, 0 ~ 0.01

- **Maximum Input Overvoltage:** ±30 V
- **Common Mode Rejection Ratio (CMRR):**

PCI-1710		PCI-1710HG	
Gain	CMRR	Gain	CMRR
0.5, 1	75dB	0.5, 1	75dB
2	80dB	10	90dB
4	84dB	100	106dB
8	84dB	1000	106dB

- **Maximum A/D data throughput:** (Hz, depending on PGIA settling time)

  PCI-1710: 100 k

  PCI-1710HG:

Gain	Speed
0.5, 1	100k
5, 10	35k
50, 100	7k
500, 1000	770

- **Accuracy:** (Depends on gain)

PCI-1710		PCI-1710HG		
Gain	Accuracy	Gain	Accuracy	Remark
0.5, 1	0.01% of FSR ±1 LSB	0.5, 1	0.01% of FSR ±1 LSB	S.E./D
2	0.02% of FSR ±1 LSB	5, 10	0.02% of FSR ±1 LSB	S.E./D
4	0.02% of FSR ±1 LSB	50, 100	0.04% of FSR ±1 LSB	D
8	0.04% of FSR ±1 LSB	500, 1000	0.08% of FSR ±1 LSB	D

**** S.E.: Single-ended D: Differential***

- **Linearity error:** ±1 LSB
- **Input impedance:** 1 GW
- **Trigger mode:** Software, on-board programmable pacer or external

Analog Output:

- **Channels:** 2
- **Resolution:** 12-bit
- **Relative accuracy:** ±1/2 LSB
- **Gain error:** ±1 LSB
- **Maximum update rate:** 100 K samples / sec (polling)
- **Slew rate:** 10 V / µs
- **Output range (software programmable):**
  Internal reference: 0 ~ +5 V @ -5 V,
  0 ~ +10 V @ -10 V
  External reference: 0 ~ +x V @ -x V (-10 ≤ x ≤ 10)

Digital Input:

- **Channels:** 16
- **Input voltage:**
  Low: 0.4 V max.
  High: 2.4 V min.
- **Input load:**
  Low: -0.2 mA @ 0.4 V
  High: 20 mA @ 2.7 V

Digital Output:

- **Channels:** 16
- **Output voltage:**
  Low: 0.4 V max. @ 8.0 mA (sink)
  High: 2.4 V min. @ -0.4 mA (source)

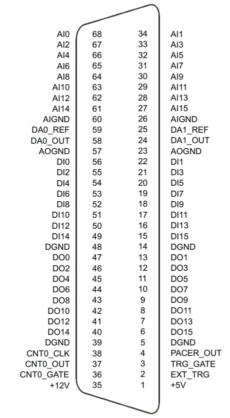
Programmable Timer/Counter

- **Counter chip:** 82C54 or equivalent
- **Counters:** 3 channels, 16 bits, 2 channels are permanently configured as a 32-bit programmable pacer; 1 channel is free for user applications
- **Input, gate:** TTL/CMOS compatible
- **Time base:**
  Channel 1: 10 MHz
  Channel 2: Takes input from output of channel 1
  Channel 0: Internal 1 MHz or external clock
  (10 MHz max.) selected by software.

General:

- **CE certified to CISPR 22 class B**
- **I/O Connector:** 68-pin SCSI-II female connector
- **Power consumption:** +5 V @ 850 mA (Typical),
  +5 V @ 1.0 A (Max.)
- **Operating temperature:** 0 ~ +60° C (32 ~ 140° F) (refer to IEC 68-2-1, 2)
- **Storage temperature:** -20 ~ +70° C (-4 ~ 158° F)
- **Operating humidity:** 5 ~ 95% RH non-condensing (refer to IEC 68-2-3)
- **Dimensions:** 175 mm (L) x 100 mm (H) (6.9" x 3.9")
- **MTBF:** over 64,770 hrs @ 25° C, grounded-fix environment

Pin Assignments



Ordering information

- ❑ **PCI-1710:** 12-bit, 100 kHz, PCI-bus Multifunction DAS Card, user's manual and driver CD-ROM. (cable not included)
- ❑ **PCI-1710HG:** 12-bit, 100 kHz, High-gain, PCI-bus Multifunction DAS Card, user's manual and driver CD-ROM. (cable not included)
- ❑ **PCLD-8710:** Wiring Terminal Board with CJC circuit (cable not included)
- ❑ **PCL-10168:** 68-pin SCSI-II cable with male connectors on both ends and special shielding for noise reduction, 1m.
- ❑ **ADAM-3968:** 68-pin SCSI-II Wiring Terminal Board for DIN-Rail Mounting

UNIVERSITI MALAYSIA PAHANG

FACULTY OF ELECTRICAL (ELECTRONIC) ENGINEERING

We certify that the project entitled "Automatic tracking object using simple robot arm" is written by Siti Khairani Binti Alias. We have examined the final copy of this project and in our opinion; it is fully adequate in terms of scope and quality for the award of the degree of Bachelor of Engineering. We herewith recommend that it be accepted in partial fulfillment of the requirements for the degree of Bachelor of Electrical and Electronic Engineering.

Examiner Signature

AUTOMATIC TRACKING OBJECT USING SIMPLE ROBOTIC ARM

SITI KHAIRANI BINTI ALIAS

Report submitted in partial fulfilment of the requirements
for the award of the degree of
Bachelor of Electric (Electronic) Engineering

Faculty of Electric (Electronic) Engineering
UNIVERSITI MALAYSIA PAHANG

OCTOBER 2010

Bachelor Final Year Project Report

Report submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Electric (Electronic).