SYAHRUL NIZAM BACHELOR OF ELECTRICAL ENGINEERING (ELECTRONICS) 2007 UMP

AUTOMATIC VEHICLE DETECTION SYSTEM

SYAHRUL NIZAM BIN ZAINI

UNIVERSITI MALAYSIA PAHANG

BORANG PENGESAHAN STATUS TESIS♦

JUDUL: **AUTOMATIC VEHICLE DETECTION SYSTEM**

SESI PENGAJIAN: **2007/2008**

Saya **SYAHRUL NIZAM BIN ZAINI (850411-04-5053)**
(HURUF BESAR)

mengaku membenarkan tesis (Sarjana Muda/~~Sarjana~~ /~~Doktor Falsafah~~)* ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Tesis adalah hakmilik Universiti Malaysia Pahang (UMP).
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. **Sila tandakan (√)

□ **SULIT** (Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

□ **TERHAD** (Mengandungi maklumat TERHAD yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan)

☑ **TIDAK TERHAD**

Disahkan oleh:

(TANDATANGAN PENULIS)

(TANDATANGAN PENYELIA)

Alamat Tetap:

BATU 20, KESANG TUA
77000 JASIN
MELAKA

NURULFADZILAH BT. HASSAN
(Nama Penyelia)

Tarikh: **20 NOVEMBER 2007**

Tarikh: : **20 NOVEMBER 2007**

CATATAN:
* Potong yang tidak berkenaan.
** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh tesis ini perlu dikelaskan sebagai atau TERHAD.
♦ Tesis dimaksudkan sebagai tesis bagi Ijazah doktor Falsafah dan Sarjana secara Penyelidikan, atau disertasi bagi pengajian secara kerja kursus dan penyelidikan, atau Laporan Projek Sarjana Muda (PSM).

"I hereby acknowledge that the scope and quality of this thesis is qualified
for the award of the Bachelor's Degree of Electrical Engineering (Electronics)"

Signature : _________________________

Name : NURULFADZILAH BT. HASSAN

Date : 23 NOVEMBER 2007

AUTOMATIC VECHICLE DETECTION SYSTEM

SYAHRUL NIZAM BIN ZAINI

A report submitted in partial fulfillment of the
requirements for the award of the degree of
Bachelor of Electrical Engineering (Electronics)

Faculty of Electrical & Electronic Engineering
University Malaysia Pahang

NOVEMBER 2007

"All the trademark and copyright use here in are property of their respective owner. References of information from other sources are quoted accordingly; otherwise the information presented in this report is solely work of the author."

Signature :________________________

Author : SYAHRUL NIZAM BIN ZAINI

Date : 23 NOVEMBER 2007

In the Name of Allah, the Most Beneficent, the Most Merciful

Special Dedication of This Great Moment to My Family
(for encouraged me in studies)
My friends, my fellow colleague
My supervisor and academic advisor
and for all those who support me in this project

"Wisdom is the most powerful"

Sincerely
Syahrul Nizam

ACKNOWLEDGEMENT

In the name of Allah S.W.T. the Most Gracious, the Most Merciful. Praise is to Allah, Lord of the Universe and Peace and Prayers be upon His final Prophet and Messenger, Muhammad S.A.W.

I would like to take this opportunity to sincerely express my highest gratitude to my supervisor, Mrs. Nurulfadzilah bt. Hassan for her guidance, ideas, advices and time during this project fulfillment. Without them, this project could not be done successfully.

My sincerely gratitude also goes to my friends for their advices, ideas and aids, especially Mohd Fadli bin Sapi'in and whose involve directly and indirectly in this project.

Lastly but not least, my specially thank to my parents, Mr. Zaini bin Osaman and Madam Hamidah bt. Hj. Hassan and my lovely brothers who had given me moral supports and always pray for my future undertakings.

ABSTRACT

This project is based on webcam, Graphical User Interface (GUI), the relationship of each part for image processing and server client connection. This system is designed to automatically & remotely determine whether parking is empty or full. The monitoring system is written using Microsoft Visual Basic Version 6 language. Edge detection is used to image reduces significantly the amount of data and filters out information that may be regarded as less relevant, preserving the important structural properties of an image. The captured image that will be compared to the default image to see the outcome if image either full or empty. This is one of image processing part that called differences. This concept cans different two images and then analysis it using Correlation coefficient concept. The data will be updated on the server that sends information on the availability of a parking space to client using internet connection. The data will be updated based on feedback of output parking system or if a client asked for the information.

ABSTRAK

Projek ini melibatkan kamera web, setiap bahagian dalam pemprosesan gambar, talian antara dengan penguna dengan pembuat servis (server) dan Antaramuka Paparan Grafik.Sistem ini direka untuk beroperasi secara automatik dan terkawal untuk mengetahui kawasan letak kereta penuh atau kosong. Sistem ini dibangunkan dengan penulisan teknologi Visual Basic versi 6.0. Konsep pengesanan sudut digunakan untuk pengurangan gambar secara bersesuaian dengan jumlah bit dalam gambar itu dengan menapis informasi yang kurang sesuai dan mengekalkan struktur gambar yang penting. Gambar yang ditangkap akan dibezakan dengan gambar asal untuk mengetahui sama ada parkir penuh atau sebaliknya. Ini salah satu konsep dalam pemprosesan gambar yang dikenali Pembezaan. Konsep ini digunakan untuk membezakan dua gambar dan analisis dengan kiraan secara konsep ”Correlation coefficient”. Data akan diperbaharui di pembuat servis (server) yang akan menghantar informasi kepada pengguna mengenai kekosongan di tempat letak kereta secara talian internet. Data juga akan diperbaharui mengikut informasi dari keluaran maklumat dari sistem atau pertanyaan dari pengguna yang inginkan maklumat baru situasi kawasan letak kereta.

TABLE OF CONTENT

LIST OF FIGURE

LIST OF TABLE

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Introduction

Nowadays, parking system is an important industrial in development country. It have many problem occurs because of a not systematic and services that produces by parking management. However, this project is one of a solution how to reduce and control parking system to become more systematic and smart.

The Automatic Vehicle Detection System is proposed to analyze parking condition, and relay information back to user. The main part of this system is an image database and graphical user interface.

The detecting system applies the analysis data and then calculates the image based on correlation coefficient concept to determine either parking if full or not. Based on the test result, the proposed system is capable of detecting and show the condition of parking system to users via internet. It is more practical and suitable with this new era of technology.

1.2 Problem statement

Currently parking management required to manually record all situation data in parking system by alert if client comment parking is full after and then put a board that say “Parking Full”. In this case client will frustrate when several problems are occur. It will give a negative feedback from client to the management.

A systematic requirement is required to solve this problem. The Automatic Vehicle Detection System is proposed to analyze parking condition remotely and relay information about available parking space back to user.

1.3 Objectives

The objective of this project is to develop a parking system that has:

i. Access control management to make system integrate between server and client smoothly.

ii. To determine whether or not a parking contains a car by using edge detection

iii. To alert users of available parking lot/space.

1.4 Scope of Project

The software will be created by using Visual Basic 6.0. What we can get from this project is the application of the project where by developing this software we will be able process the image. This will make us more understand more about image processing as we know that there is variety of application image processing.

This system also had been developed based on parking system requirement. It is important to make sure the system developed will meets their requirement. The system will only focus on the image captured using a webcam, enchantment image to edge detection and produce analysis based on comparison data. There will be two users who will use this system. It is the administrator and client. Client used it to acknowledge the parking situation via internet. While administrator is specialize to give information either parking if full or available.

1.5 Methodology

Figure 1.1 show the flow chart of the system design for this project where the first step is literature review. This part is the most important step before doing any project to give some information about the project. For this project, the information gathered from surfing internet, books and writing materials and a brief discussion with supervisor and an expert. From this literature review, we will know about the software that need for this project.

After getting all of the information about the project, the next step is to design and develop automatic vehicle detection system. For this project we need to analyze parking condition remotely, and relay information about available parking space back to user.

The last step is integrate both of the system and test whether the system archive the goal or not and make an analysis of the project result.

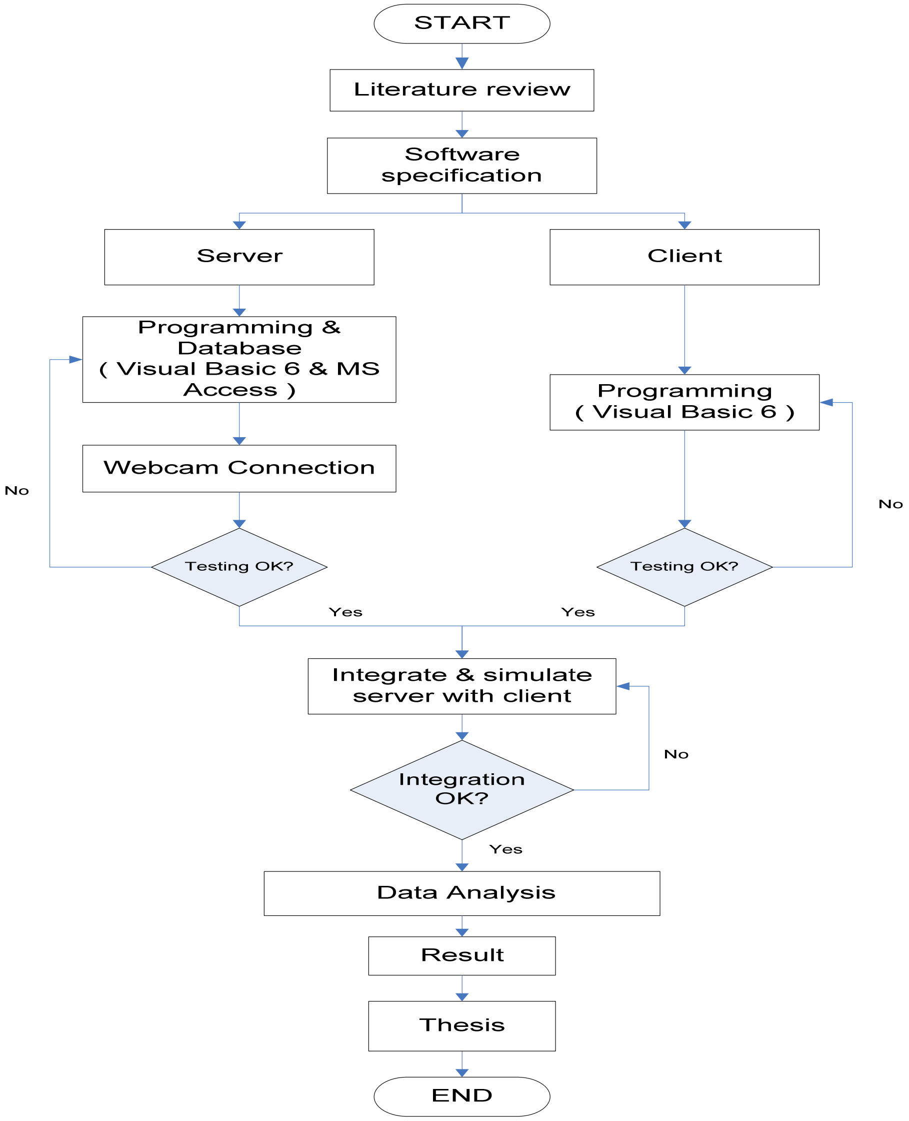


Figure 1.1 Flow Chart Progresses of PSM1 and PSM2

1.6 Thesis Outline

The thesis is organized as follows:

Chapter 1: This chapter explained about project background, objective, scope, methodology and also the problem statement.

Chapter 2: Describe about webcam, server, data acquisition and the relationship of each part for image processing

Chapter 3: Discuss a detail of the design for Automatic Vehicle Detection System. Meanwhile, represent the main part of the project, edge detection.

Chapter 4: Describe about the experimental results and include two part which are server page and client page

Chapter 5: presents the overall conclusion for this thesis and a few suggestion and recommendation for future work.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter explains the background of webcam, server and the relationship of each part for image processing. This chapter also explains the overview of image processing for automatic vehicle detection system project. In this chapter, overview everything based on literature review about all concept including in this project.

2.2 Data Acquisition

Data acquisition is the sampling of the real world to generate data that can be manipulated by a computer. Sometimes abbreviated DAQ or DAS, data acquisition typically involves acquisition of signals and waveforms and processing the signals to obtain desired information [J. Maps, Jul 1993].

Acquired data is displayed, analyzed, and stored on a computer, either using vendor supplied software, or custom displays and control can be developed using various text-based programming languages such as BASIC, C, Fortran, Java, Lisp, Pascal. LabVIEW offers a graphical programming environment optimized for data acquisition. MATLAB provides a programming language but also built-in graphical tools and libraries for data acquisition and analysis [Charles, 1990].

2.3.3.1 Webcam

Started in 1991, the first webcam was pointed at the Trojan room coffee pot in the computer science department of Cambridge University. This webcam is now defunct, as it was finally switched off on August 22, 2001. The final image captured by the camera can still be viewed at the webcam's homepage [Oliver, 2004].

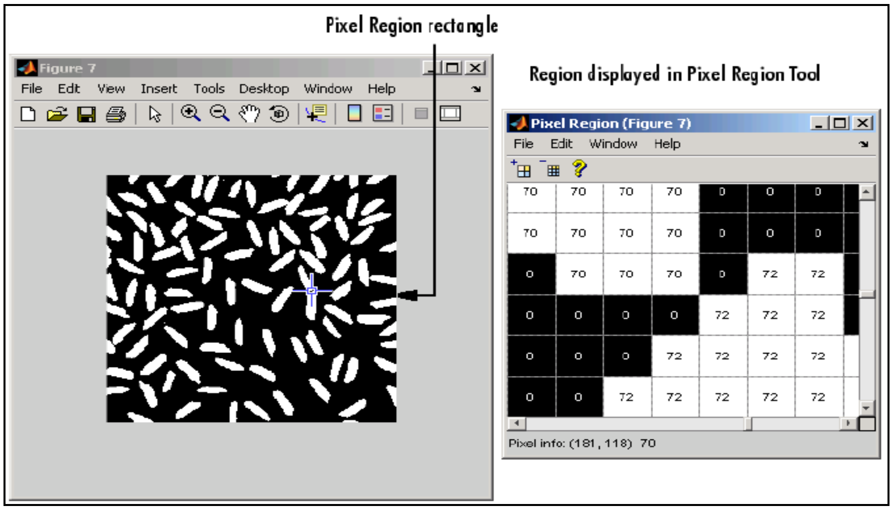
Webcams connected to PCs can act as web-accessible cameras with certain software; the software uploads pictures to a server, from which can produce an input to system. Usually, this kind of software is programmed to work with almost every webcam. This software can be configured in many ways, and will often include options for image size and quality, overlaying logos, and time stamping images.

In this project, the webcam is eye of this system to capture a picture on parking park and sends back to system to analysis it. The webcam will capture an image after 30 second that set in system. The image will save in bitmap file. The webcam use in this system is shown in Figure 2.1.

Figure 2.1 Webcam

2.4 Image Processing

Image processing is any form of information processing for which the input is an image, such as photographs or frames of video; the output is not necessarily an image, but can be for instance a set of features of the image. Most image-processing techniques involve treating the image as a two-dimensional signal and applying standard signal-processing techniques


Pixel Region rectangle
Region displayed in Pixel Region Tool
Figure 7
File Edt View Insert Tools Desktop Window Help
Pixel Region (Figure 7)
File Edit Window Help

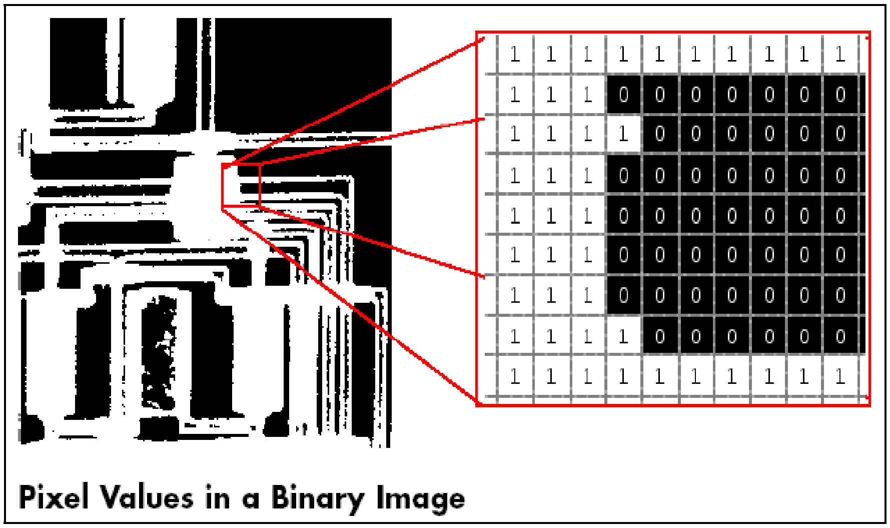
70	70	70	70	0	0	0
70	70	70	70	0	0	0
0	70	70	70	0	72	72
0	0	0	0	72	72	72
0	0	0	72	72	72	72
0	0	72	72	72	72	72

Pixel info: (181 , 118) 70

Figure 2.2 Image Processing

An image defined in the "real world" is considered to be a function of two real variables, for example, *a*(*x*,*y*) with *a* as the amplitude (e.g. brightness) of the image at the *real* coordinate position (*x*,*y*). An image may be considered to contain sub-images sometimes referred to as *regions-of-interest*, *ROIs*, or simply *regions*. This concept reflects the fact that images frequently contain collections of objects each of which can be the basis for a region. Figure 2.2(a) show the pixel on image that converts in dot matrix. In a sophisticated image processing system it should be possible to apply specific image processing operations to selected regions. Thus one part of an image (region) might be processed to suppress motion blur while another part might be processed to improve color rendition [Young, 1995].

The amplitudes of a given image will almost always be either real numbers or integer numbers. Figure 2.3 show that pixel value in a binary image. The latter is usually a result of a quantization process that converts a continuous range (say, between 0 and 100%) to a discrete number of levels. In certain image-forming processes, however, the signal may involve photon counting which implies that the amplitude would be inherently quantized. In other image forming procedures, such as magnetic resonance imaging, the direct physical measurement yields a complex number in the form of a real magnitude and a real phase [Scott, 2005].



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

Figure 2.3 Pixel values in a binary image

In image processing, must consider the amplitudes as real's or integers unless otherwise indicated. Visual Basic, the visual-development environment that prefer to work in, often falls short in supporting these requirements, particularly when it comes to fast image plotting. However, it is possible to achieve efficient image processing by improving disk access and image plotting speeds using the techniques.

2.4.1 Edge Detection

The goal of edge detection is to mark the points in a digital image at which the luminous intensity changes sharply. Sharp changes in image properties usually reflect important events and changes in properties of the world. These include:

(i) Discontinuities in depth
(ii) Discontinuities in surface orientation
(iii) Changes in material properties
(iv) Variations in scene illumination

Edge detection of an image reduces significantly the amount of data and filters out information that may be regarded as less relevant, preserving the important structural properties of an image. There are many methods for edge detection, but most of them can be grouped into two categories, search-based and zero-crossing based. The search-based methods detect edges by looking for maxima and minima in the first derivative of the image, usually local directional maxima of the gradient magnitude. The zero-crossing based methods search for zero crossings in the second derivative of the image in order to find edges, usually the zero-crossings of the Laplacian or the zero-crossings of a non-linear differential expression [Qiu, 2005]

Most edge detection methods work on the assumption that an edge occurs where there is a discontinuity in the intensity function or a very steep intensity gradient in the image. Using this assumption, if we take the derivative of the intensity values across the image and find points where the derivative is a maximum, we will have marked our edges.

A number of researchers have considered the problem of measuring edge detector performance. In fact, it is difficult since we don't really know what the underlying features are that we wish to detect. However, if we assume that they are step edges corrupted by Gaussian noise, then some criteria can be set for evaluating performance. Such criteria are usually the following:

- the probability of false edges;
- the probability of missing edges;
- the error in estimating the edge angle;
- the mean square distance of the edge estimate from the true edge; and
- The algorithm's tolerance to distorted edges and other features such as corners and junctions.

Currently, the canny operator (or variations of this operator) is most commonly used edge detection method. A large number of edge detection operators have been published but so far none has shown significant advantages over the Canny-type operators in general situations. In his original work, Canny studied the problem of designing an optimal pre-smoothing filter for edge detection, and then showed that this filter could be well approximated by a first-order Gaussian derivative kernel. Canny also introduced the notion of non-maximum suppression, which means that edges are defined as points where the gradient magnitude assumes a maximum in the gradient direction [Canny, 2007].

2.4.2 Threshold Image

Threshold requires a suitable reference image and the selection of an appropriate detection threshold. Several threshold selection methods are investigated, and an algorithm based on hysteresis thresholding is shown to give acceptably good results over a number of test image sets.

To performing the automatic thresholding of difference images is to assume particular distribution models for the difference of image samples and the noise. The image was thresholded at multiple intensities, and the connectivity value of each calculated. The threshold was selected from an intensity range that produced a stable set of connectivity values. Rather than measuring connectivity, the number of regions may be more appropriate [Gonzalez & Else, 2002]

2.4.3 Pattern Matching

In computer science, pattern matching is the act of checking for the presence of the constituents of a given pattern. In contrast to pattern recognition, the pattern is rigidly specified. Such a pattern concerns conventionally either sequences or tree structures. Pattern matching is used to test whether things have a desired structure, to find relevant structure, to retrieve the aligning parts, and to substitute the matching part with something else. Sequence (or specifically text string) patterns are often described using regular expressions (i.e. backtracking) and matched using respective algorithms. Sequences can also be seen as trees branching for each element into the respective element and the rest of the sequence, or as trees that immediately branch into all elements [Pierre, 2004].

Tree patterns can be used in programming languages as a general tool to process data based on its structure. Some functional programming languages such as Haskell, ML and the symbolic mathematics language Mathematical have a special syntax for expressing tree patterns and a language construct for conditional execution and value retrieval based on it. For simplicity and efficiency reasons, these tree patterns lack some features that are available in regular expressions. Often it is possible to give alternative patterns that are tried one by one, which yields a powerful conditional programming construct. Pattern matching can benefit from guards [Pierre, 2004].

Term rewriting languages rely on pattern matching for the fundamental way a program evaluates into a result. Pattern matching benefits most when the underlying data structures are as simple and flexible as possible. This is especially the case in languages with a strong symbolic flavor. In symbolic programming languages, patterns are the same kind of data type as everything else, and can therefore be fed in as arguments to functions [Pierre, 2004].

2.5 Database

Database can be defined as a structured collection of records or data that is stored in a computer so that a program can consult it to answer queries. The records retrieved in answer to queries become information that can be used to make decisions [Thomas & Else, 2004].

The central concept of a database is that of a collection of records, or pieces of information. Typically, for a given database, there is a structural description of the type of facts held in that database: this description is known as a schema. The schema describes the objects that are represented in the database, and the relationships among them. There are a number of different ways of organizing a schema, that is, of modeling the database structure: these are known as database models [Thomas & Else, 2004].

Databases are used in many applications, spanning virtually the entire range of computer software. Databases are the preferred method of storage for large multiuser applications, where coordination between many users is needed. Even individual users find them convenient, and many electronic mail programs and personal organizers are based on standard database technology. Software database drivers are available for most database platforms so that application software can use a common application programming interface (API) to retrieve the information stored in a database. Two commonly used database APIs are JDBC and ODBC [Thomas & Else, 2004].

2.6 Microsoft Visual Basic

Visual Basic was designed to be easy to learn and use. The language not only allows programmers to easily create simple GUI applications, but also has the flexibility to develop fairly complex applications as well. Programming in visual basic is a combination of visually arranging components or controls on a form, specifying attributes and actions of those components, and writing additional lines of code for more functionality. Since default attributes and actions are defined for the components, a simple program can be created without the programmer having to write many lines of code. Performance problems were experienced by earlier versions, but with faster computers and native code compilation this has become less of an issue [Michael, 1998].

2.7 Microsoft Access

Access allows relatively quick development because all database tables, queries, forms, and reports are stored in the database. For query development, Access utilizes the Query Design Grid, a graphical user interface that allows users to create queries without knowledge of the SQL programming language. In the Query Design Grid, users can "show" the source tables of the query and select the fields they want returned by clicking and dragging them into the grid. Joins can be created by clicking and dragging fields in tables to fields in other tables. Access allows users to view and manipulate the SQL code if desired [Michael, 1998].

Microsoft Access can be applied to small projects but scales poorly to larger projects involving multiple concurrent users because it is a desktop application, not a true client-server database. When a Microsoft Access database is shared by multiple concurrent users, processing speed suffers. The effect is dramatic when there are more than a few users or if the processing demands of any of the users are high. Access includes an Upsizing Wizard that allows users to upsize their database to Microsoft SQL Server if they want to move to a true client-server database. Since all database queries, forms, and reports are stored in the database, and in keeping with the ideals of the relational model, there is no possibility of making a physically structured hierarchy with them[Michael, 1998].

One design technique is to divide an Access application between data and programs. One database should contain only tables and relationships, while another would have all programs, forms, reports and queries, and links to the first database tables. Unfortunately, Access allows no relative paths when linking, so the development environment should have the same path as the production environment. This technique also allows the developer to divide the application among different files, so some structure is possible [Michael, 1998]

CHAPTER 3

SYSTEM DESIGN

3.1 Introduction

In developing a project, system designing is one of the most important elements to be considered to make sure that the development of the project is smooth and get the expected result. A good system design can described the structure or the flow of the project where by it can be the guideline in managing it. It is also to avoid the project to alter course from the objectives that have been stated or in other words the project follow the guideline based on the objectives

3.2 Block Diagram of the system

This project aims to develop an Image Processing of an Automatic Vehicle Detection system to detect whether parking if full or available. Webcam will be the heart of the system and it will capture an image and then give the data to server to analyze it. The main target of this project is to give information to client either parking if full or available via internet. Figure 3.1 shows the system design of automatic vehicle detection system.

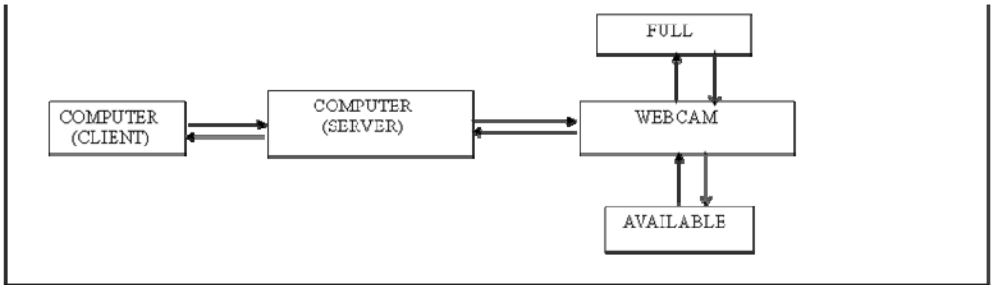


Figure 3.1 System Design

3.3 Design of the System Architecture of Automatic Vehicle Detection System

Figure 3.2 illustrates the relationship between the GUI, the image database, the Image processing concept and the server-client connection. The GUI is the user interface platform of monitoring system. Whereas, the database is the storage system for the un-processed image. Then, the image processing concept is the engine that runs the system. It is consists of an edge detection, filtering, thresholding and display the image analysis. Server – client connection is to give information if parking is empty or full.

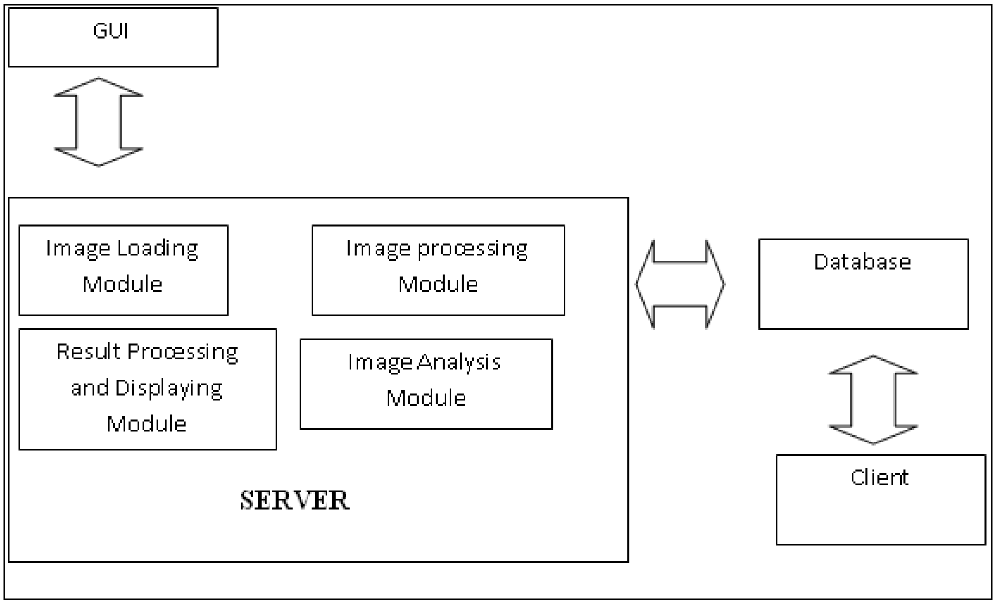


Figure 3.2 The architecture of the monitoring system

3.4 Server

This part discuss about the flow chart for Server page. This page are develop to allow administrator organize the vehicle availability and key in the information to users.

3.4.1 Administrator Login

Figure 3.3 shows the flow chart to login server page. To login this page the administrator need to enter the user name and the password for safety. If the username and password is valid, it will go to the system and if not the administrator need to login again.

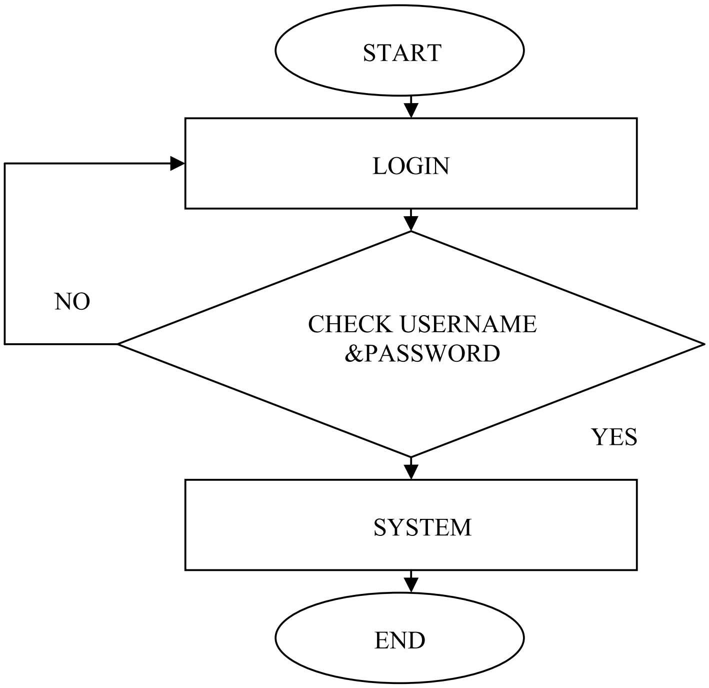


Figure 3.3 Flow Chart for Server Login

3.4.2 Automatic Vehicle Detection System Flow Chart

Figure 3.4 shows the flow chart of destination Automatic Vehicle Detection System. This system is enable administrator to start the program that running to capture an image, convert to edge detection, analysis data and give the result. The server is the main part of this project.

This part described more on image processing process applies in this system. Edge detection is use to enchantments image that improve the quality of image. Image analysis is the key point to different image and get an output after calculate image by using coefficient correlation concept. The data will show the parking situation and save in database.

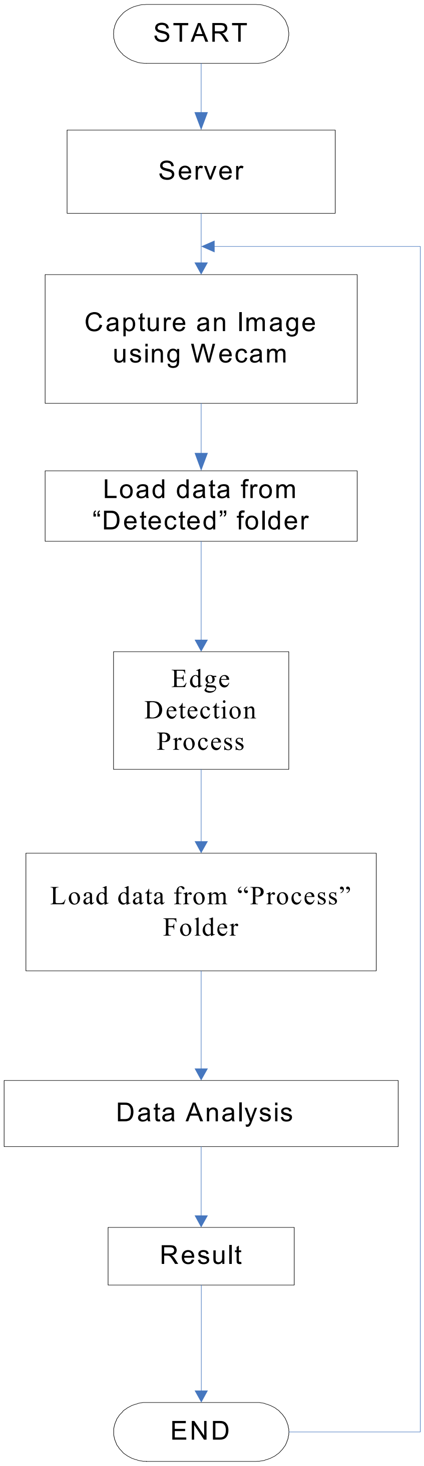

START
Server
Capture an Image using Wecam
Load data from "Detected" folder
Edge Detection Process
Load data from "Process" Folder
Data Analysis
Result
END

Figure 3.4 Flow chart for Automatic Vehicle Detection

3.4.3 Client

In this part, the system will integrate between server and client via internet. The client system depends on output (result) on the availability parking space. This system is links a form to a database and allows system to access the fields in the tables making up the database.

Figure 3.5 shows the flow chart of client in this system. This system is enable client to view the availability of parking by enter the date. The system will match the date with database and give the feedback.

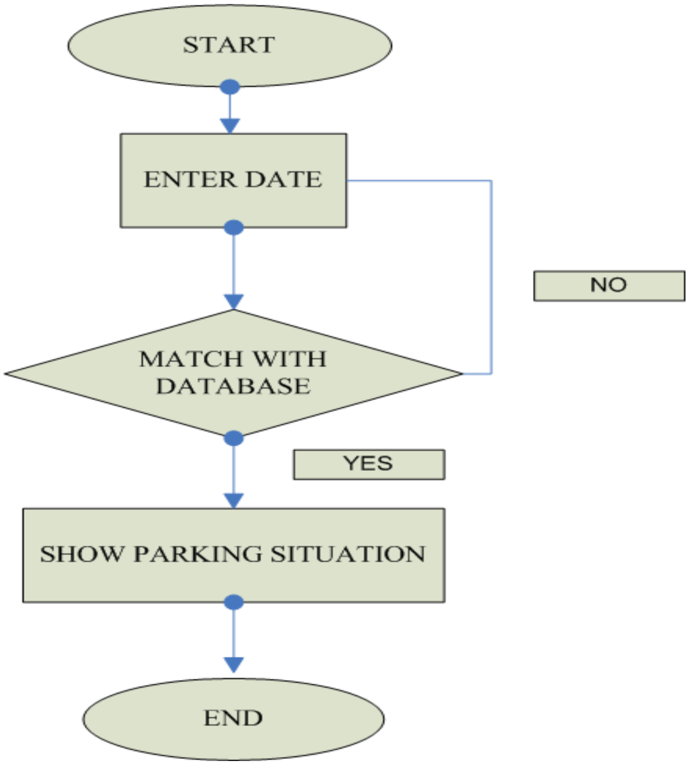
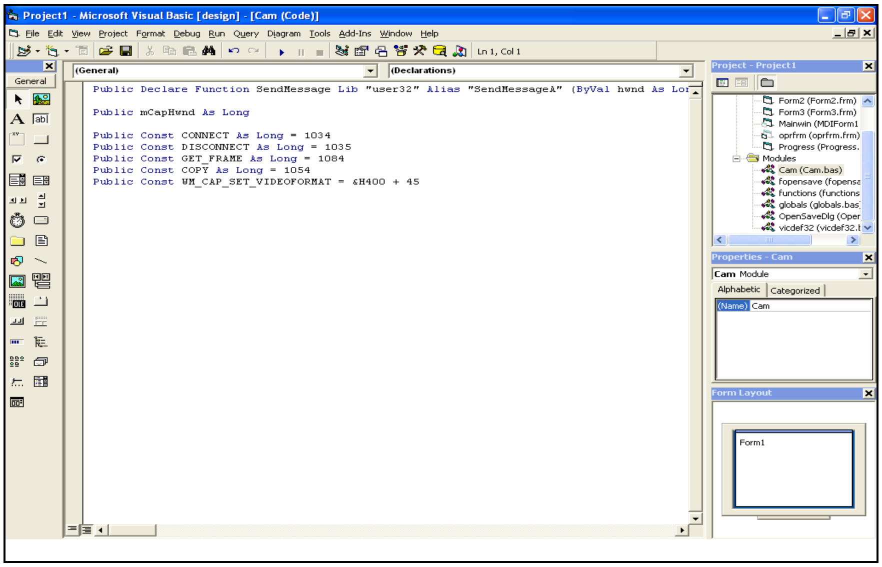


Figure 3.5 Flow Chart for Client

3.5 Build Webcam Connection

In this system, webcam connection is use USB connector. In Visual Basic 6.0, must declare the module that can give a signal to webcam to function with system. In this system, the module saves as Cam.bus. Moreover, the main page must use a coding that connects this module and then the webcam can function smoothly with the Automatic Vehicle Detection System. The command is in show in Figure 3.6



```
Public Declare Function SendMessage Lib "user32" Alias "SendMessageA" (ByVal hwnd As Lor

Public mCapHwnd As Long

Public Const CONNECT As Long = 1034
Public Const DISCONNECT As Long = 1035
Public Const GET_FRAME As Long = 1084
Public Const COPY As Long = 1054
Public Const WM_CAP_SET_VIDEOFORMAT = &H400 + 45
```

Figure 3.6 Command in webcam connection

3.6 Image Processing Program

Image processing is the capability of software to manipulate images that are represented by a sequence of numeric values. This section will give a key process in this system. There are edge detection, threshold and image analysis. Each part in image processing will discuss in more detail about how system work.

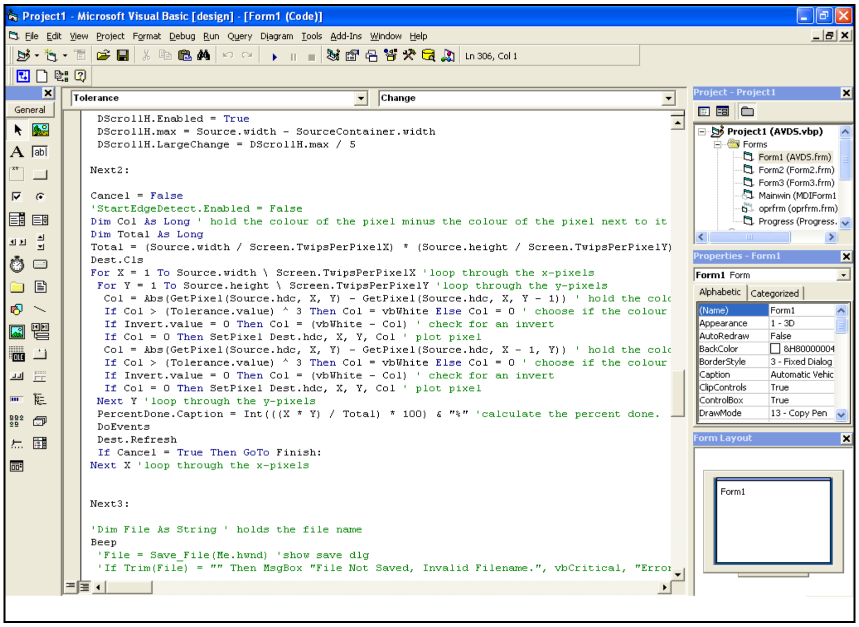
3.6.1 Edge Detection

This is a key of the system. It is create to convert a color image to white and black. If there were sensor with infinitely small footprints and zero-width point spread functions, an edge would be recorded between pixels within in an image. Theoretically, edge detection of an image reduces significantly the amount of data and filters out information that may be regarded as less relevant, preserving the important structural properties of an image.

First, create a module, naming it like named function.bas, then declare string arrays in that module under the *general* attribute. The string declarations in this program are *Global Datarray As String* and *Workarray As String,* which define *Datarray* and *Workarray* as globally recognized string arrays.

In this system string is used because basic-type programs use them efficiently. Note that a specific size is not needed for the array definition. This provides some programming flexibility if want to enter files with different formats and lengths.

The command in this system is inside a code window as shown in Figure 3.7. This proposed system also use the image file ".bmp" in this system because this image format is platform dependence, supports a simple's lossless compression and stored uncompressed. RGB is a 24-bit methodology where color is specified in terms of red, green, and blue values ranging from 0 to 255. This RGB use to convert to edge detection.



```
 DScrollH.Enabled = True
 DScrollH.max = Source.width - SourceContainer.width
 DScrollH.LargeChange = DScrollH.max / 5

Next2:

Cancel = False
'StartEdgeDetect.Enabled = False
Dim Col As Long ' hold the colour of the pixel minus the colour of the pixel next to it
Dim Total As Long
Total = (Source.width / Screen.TwipsPerPixelX) * (Source.height / Screen.TwipsPerPixelY)
Dest.Cls
For X = 1 To Source.width \ Screen.TwipsPerPixelX 'loop through the x-pixels
 For Y = 1 To Source.height \ Screen.TwipsPerPixelY 'loop through the y-pixels
  Col = Abs(GetPixel(Source.hdc, X, Y) - GetPixel(Source.hdc, X, Y - 1)) ' hold the colc
  If Col > (Tolerance.value) ^ 3 Then Col = vbWhite Else Col = 0 ' choose if the colour
  If Invert.value = 0 Then Col = (vbWhite - Col) ' check for an invert
  If Col = 0 Then SetPixel Dest.hdc, X, Y, Col ' plot pixel
  Col = Abs(GetPixel(Source.hdc, X, Y) - GetPixel(Source.hdc, X - 1, Y)) ' hold the colc
  If Col > (Tolerance.value) ^ 3 Then Col = vbWhite Else Col = 0 ' choose if the colour
  If Invert.value = 0 Then Col = (vbWhite - Col) ' check for an invert
  If Col = 0 Then SetPixel Dest.hdc, X, Y, Col ' plot pixel
 Next Y 'loop through the y-pixels
 PercentDone.Caption = Int(((X * Y) / Total) * 100) & "%" 'calculate the percent done.
 DoEvents
 Dest.Refresh
 If Cancel = True Then GoTo Finish:
Next X 'loop through the x-pixels


Next3:

'Dim File As String ' holds the file name
Beep
 'File = Save_File(Me.hwnd) 'show save dlg
 'If Trim(File) = "" Then MsgBox "File Not Saved, Invalid Filename.", vbCritical, "Error
```

(Name)	Form1
Appearance	1 - 3D
AutoRedraw	False
BackColor	□ &H80000004
BorderStyle	3 - Fixed Dialog
Caption	Automatic Vehic
ClipControls	True
ControlBox	True
DrawMode	13 - Copy Pen

Figure 3.7 A set of command of edge detection

3.6.2 Threshold

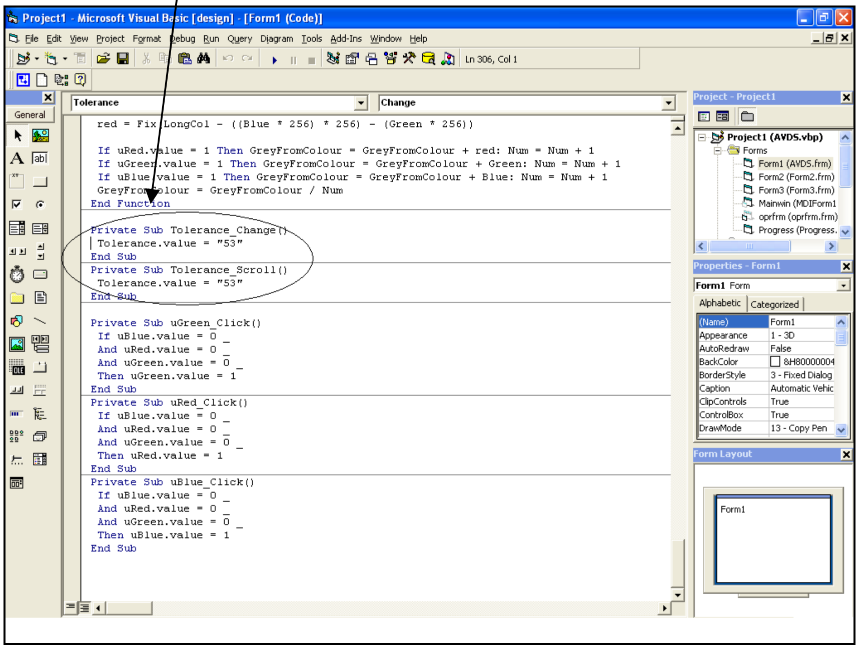
Once after calculated derivative, the next stage is to apply a threshold, to determine where the result suggests an edge to be present. The lower the threshold, the more lines will be detected, and the results become increasingly susceptible to noise, and also to picking out irrelevant features from the image. Conversely a high threshold may miss subtle lines, or segmented lines.

A commonly used compromise is thresholding with hysteresis. This method uses multiple thresholds to find edges. Begin by using the upper threshold to find the start of a line. Once start point, this system will trace the edge's path through the image pixel by pixel and marking an edge. The system stops marking edge only when the value falls below lower threshold. This approach makes the assumption that edges are likely to be in continuous lines, and allows following a faint section of an edge that have previously seen, without meaning that every noisy pixel in the image is marked down as an edge.

Changing the lower box's value from 255 to a lower amount will constrict the threshold range. In the example above, all pixel values with intensity lower than 107 and all those with intensity 141 will be forced black. All those in between will be forced white. The white pixels will be 30% of the pixels in the image.

In this system, the threshold is set as 53/255. The image will mark edge more efficient in this system and preserving the important structural properties of an image. The command is show in Figure 3.8.

Threshold is set “53”



```
 red = Fix(LongCol - ((Blue * 256) * 256) - (Green * 256))

 If uRed.value = 1 Then GreyFromColour = GreyFromColour + red: Num = Num + 1
 If uGreen.value = 1 Then GreyFromColour = GreyFromColour + Green: Num = Num + 1
 If uBlue.value = 1 Then GreyFromColour = GreyFromColour + Blue: Num = Num + 1
 GreyFromColour = GreyFromColour / Num
End Function

Private Sub Tolerance_Change()
 Tolerance.value = "53"
End Sub
Private Sub Tolerance_Scroll()
 Tolerance.value = "53"
End Sub

Private Sub uGreen_Click()
 If uBlue.value = 0 _
 And uRed.value = 0 _
 And uGreen.value = 0 _
 Then uGreen.value = 1
End Sub
Private Sub uRed_Click()
 If uBlue.value = 0 _
 And uRed.value = 0 _
 And uGreen.value = 0 _
 Then uRed.value = 1
End Sub
Private Sub uBlue_Click()
 If uBlue.value = 0 _
 And uRed.value = 0 _
 And uGreen.value = 0 _
 Then uBlue.value = 1
End Sub
```

(Name)	Form1
Appearance	1 - 3D
AutoRedraw	False
BackColor	□ &H80000004
BorderStyle	3 - Fixed Dialog
Caption	Automatic Vehic
ClipControls	True
ControlBox	True
DrawMode	13 - Copy Pen

Figure 3.8 Command of threshold

3.6.3 Image Analysis

Image analysis is concerned with the extraction of quantitative information from image captured in digital form. Image analysis tasks can be as simple as reading sophisticated as identifying a picture by calculated the dots pixel in image. A wide range of data can be acquired using image analysis. Also, monographs on image analysis technique [e.g., Russ, 1999] are oriented towards medicine, biology or material science.

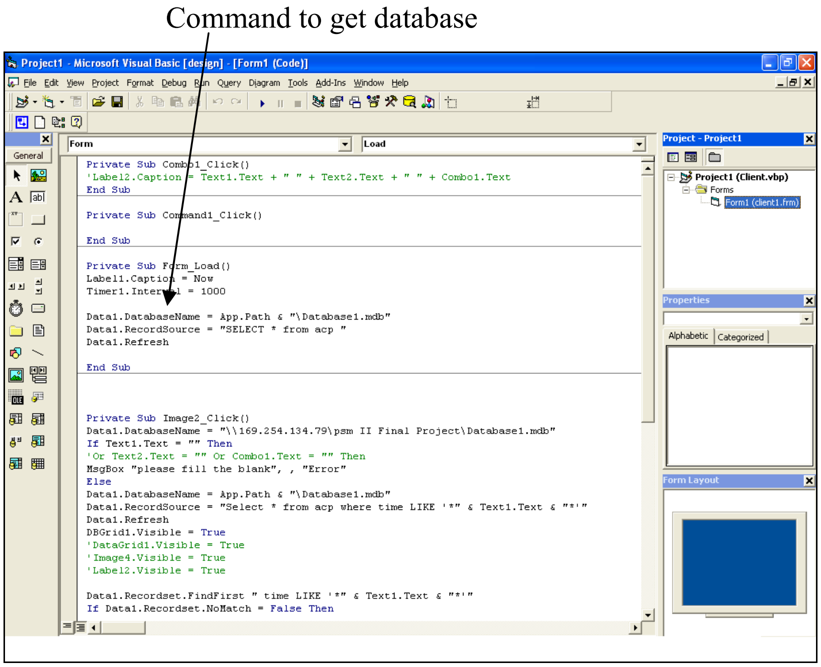
The representatively of the measurement made on images should always be kept in mind because the pixels are samples of an image, images are samples of the sample under investigation, the sample under investigation are sample of the sediment of interest.

Each step of image processing should be carefully tested using sets of calibration images or test images. In this system, the images can be analysis and get the result after do a correlation coefficient algorithm to calculate the pixels.

3.7 Parking Database

In Automatic Vehicle Detection System, the database platform uses a Microsoft Access. Access includes an Upsizing Wizard that allows users to upsize their database to Microsoft SQL Server if they want to move to a true client-server database. One design technique is to divide an Access application between data and programs. One database should contain only tables and relationships, while another would have all programs, forms, reports and queries, and links to the first database tables.

In this system the database save the result and send back to client. The database will send based on matching date on that database only. The system on visual basic will integrate with Microsoft Access with use data control to connect this system to work efficiently. Figure 3.9 will show the command to match database that type on client page.


Command to get database

Project1 - Microsoft Visual Basic [design] - [Form1 (Code)]
File Edit View Project Format Debug Run Query Diagram Tools Add-Ins Window Help
General
Form
Load
Project - Project1
Project1 (Client.vbp)
Forms
Form1 (client1.frm)
Properties
Alphabetic Categorized
Form Layout

```
Private Sub Combo1_Click()
'Label2.Caption = Text1.Text + " " + Text2.Text + " " + Combo1.Text
End Sub

Private Sub Command1_Click()

End Sub

Private Sub Form_Load()
Label1.Caption = Now
Timer1.Interval = 1000

Data1.DatabaseName = App.Path & "\Database1.mdb"
Data1.RecordSource = "SELECT * from acp "
Data1.Refresh

End Sub



Private Sub Image2_Click()
Data1.DatabaseName = "\\169.254.134.79\psm II Final Project\Database1.mdb"
If Text1.Text = "" Then
'Or Text2.Text = "" Or Combo1.Text = "" Then
MsgBox "please fill the blank", , "Error"
Else
Data1.DatabaseName = App.Path & "\Database1.mdb"
Data1.RecordSource = "Select * from acp where time LIKE '*" & Text1.Text & "*'"
Data1.Refresh
DBGrid1.Visible = True
'DataGrid1.Visible = True
'Image4.Visible = True
'Label2.Visible = True

Data1.Recordset.FindFirst " time LIKE '*" & Text1.Text & "*'"
If Data1.Recordset.NoMatch = False Then
```

Figure 3.9 Command to get database

3.8 Client – Server

The client communicated with a name server to translate the server name into an **IP Address**, which it uses to connect to the server computer. The browser then formed a connection to the server at that IP address on port 80. Connection between these two computers is use null modem cable. A T1 line can handle approximately 1.5 million bits per second, while a normal phone line using a modem can typically handle 30,000 to 50,000 bits per second.

To keep this entire computer integrate, each computer on the Internet is assigned a unique address called an IP address. IP stands for Internet protocol, and these addresses are 32-bit numbers, normally expressed as four "octets" in a "dotted decimal number". A set of servers called domain name servers (DNS) maps the human-readable names to the IP addresses These servers are simple databases that map names to IP addresses, and they are distributed all over the Internet.

Each instance of the client software can send data requests to one or more connected servers. In turn, the servers can accept these requests, process them, and return the requested information to the client. Although this concept can be applied for a variety of reasons to many different kinds of applications, the architecture remains fundamentally the same. The integrate connection between server and client is shown in Figure 3.10

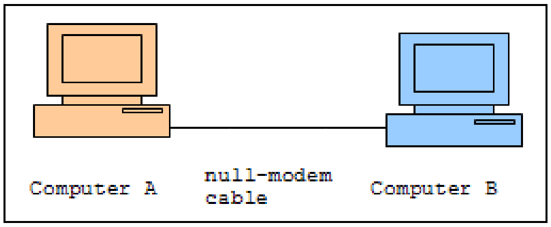


Figure 3.10 Interface design

3.9 Development Tools

In order for Automatic Vehicle Detection System to work accordingly, there are a few specific hardware and software requirement that have to be met. Below is the system requirement that have been chosen for develop Automatic Vehicle Detection System:-

3.9.1 Hardware and Software Requirements

The hardware used to develop the system is as listed in Table 3.1

Table 3.1 Table Hardware Requirements

Item	**Quantity**	**Minimum Requirements**	**Purpose**
Personal computer	1	Intel(R) Pentium(R) M 1.60 GHz Hard Disk 40GB RAM 256MB	For documentation and development.
Printer	1	Canon 1600sp	Documentation, Review
CD-R	3	700MB	Backup file.

The software required to develop the system are as listed in Table 3.2.

Table 3.2 Table of Software Requirements

Item	**Name**	**Purposes**
Operating System	Microsoft Windows XP Professional	Operating system that are used when developing whole project including documentation.
Software	Microsoft Visual Basic 6.0	Interface design & coding.
	Microsoft Access	Database platform
	Microsoft Office Word 2003	Document the thesis.
	Microsoft Office Project 2003	Document the Project Gantt Chart.
	Microsoft Office PowerPoint 2003	Presentation.
	Adobe Photoshop 7.0	To edit image.

3.10 Summary

As conclusion, this chapter defines the system design phase by phase in this phase will described about an idea to design the system that develop in this system before. Beside that in this chapter also discus about all of the software needed to develop this project. For more details, can review coding in Appendix B and C. it includes the coding that using in Automatic Vehicle Detection System.

CHAPTER 4

RESULT

4.1 Introduction

This chapter will discuss all o the result from this project. There include two part which are server page and client page. A system that changes an image to edge detection and then analysis it to determine whether parking full or available. The server sends information to client to let it know the parking park availability.

4.2 Graphical User Interface (GUI)

The GUI is an interactive user-friendly interface. It enables the user to immediately use the monitoring system without having to read detailed instruction manual because each button has a function that clearly defined. For instance, a command button I designed to load an image, would be written with “Load” on it. The GUI developed for monitoring system is shown in Figure 4.1. Its components and their functions are tabulated in Table 4.1. The GUI is developed using Visual Basic 6 language. The functions of the components of the GUI are described below.

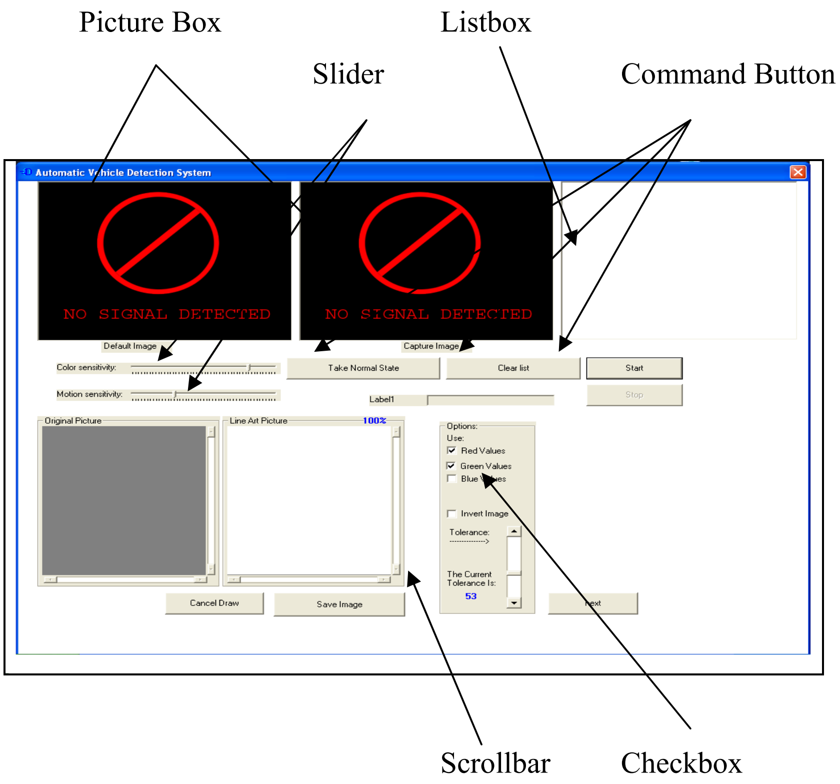


Figure 4.1 The GUI of the proposed system

Table 4.1 The selected GUI components and their function

Components	Function
Form window	To provide enough space to place all control tools inside it
PictureBox Control	To place original image and to draw color lines surrounding the image and enchantments it to edge detection.
Listbox	To list down an save image that capture from webcam
Start Button	To start the system function
Next Button	To next form that analysis data
Checkbox	To convert RGB to white during edge detection
Scrollbar	To get a suitable threshold during edge detection function
Timer Control	To give a time delay during capture an image from webcam
Clear list Button	To clear a list on the list box.

4.3 Server Page

Figure 4.2 shows the login page for administrator. The server page must have a password to avoid other people access the system. This means only administrator that have a valid username and password can login, key in and manage the system.

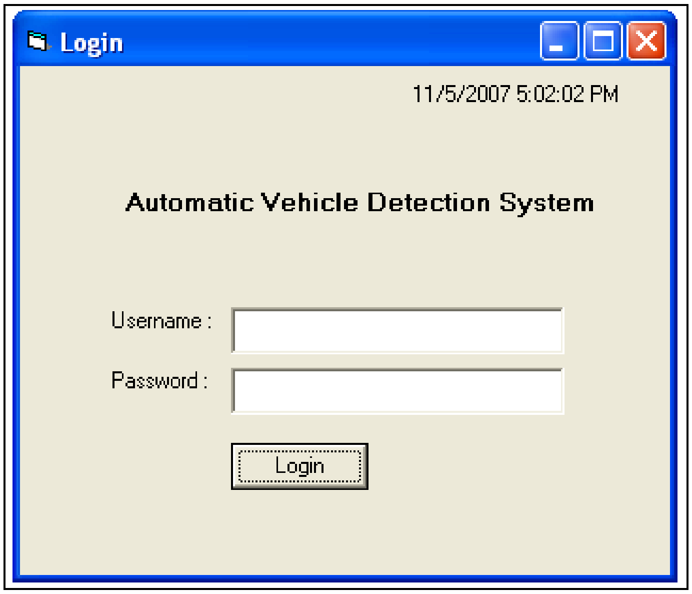
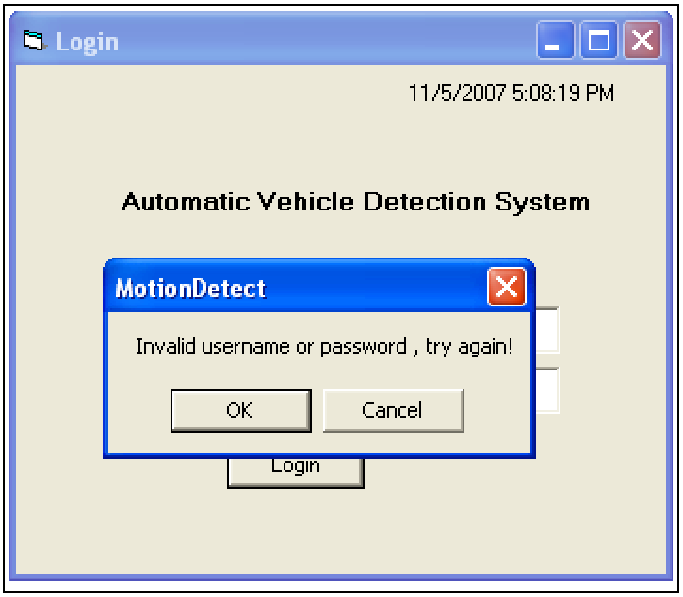


Figure 4.2 Server Login Page

This login page can be access through the Automatic Vehicle Detection System. Only valid username are allowed to enter the system. If the username or password is not valid the text "Invalid Username or Password, try again!" will appear as shown in Figure 4.3



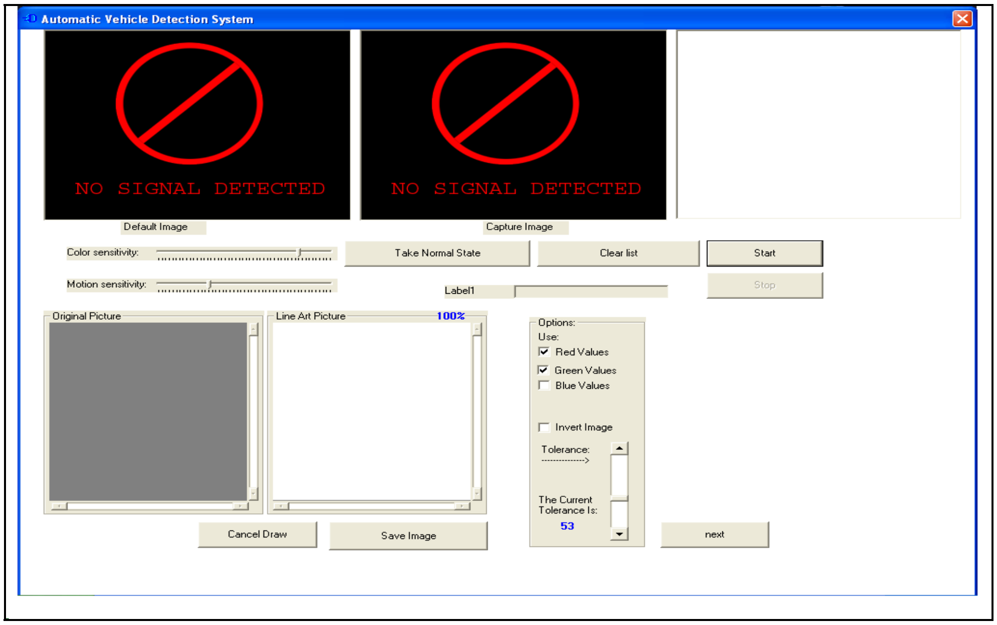
11/5/2007 5:08:19 PM

Automatic Vehicle Detection System

Invalid username or password , try again!

Figure 4.3 Access Denied

If the password is valid and match to the database it will go another page that the main menu of the system as shown in Figure 4.4.



100%

Options:
Use:
☑ Red Values
☑ Green Values
□ Blue Values

□ Invert Image

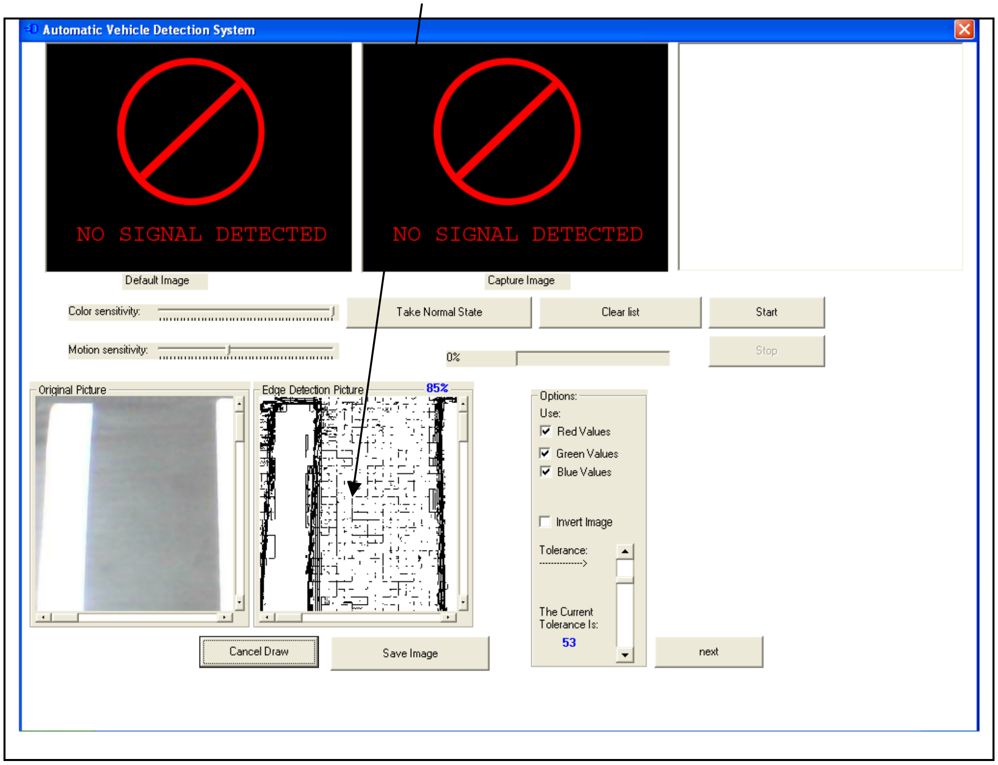
The Current Tolerance Is:
53

Figure 4.4 Main Menu of the System

In this page server have seven options either to click Start, clear list, take normal state, save image, cancel draw or next to go next form the system. If the administrator clicks at Start button the system will automatically function and capture an image shows the capture image as shown in Figure 4.5.

After that, an image will save automatically in “Detected” folder. Then, administrator must load that picture (newest capture image) and the image will convert to edge detection. A sample of output of the system is show in Figure 4.6(a) and Figure 4.6(b).

Images convert to edge detection



0%

85%

Options:
Use:
☑ Red Values
☑ Green Values
☑ Blue Values
☐ Invert Image
Tolerance:
------------->
The Current Tolerance Is:
53

Figure 4.5 An Edge Detection function

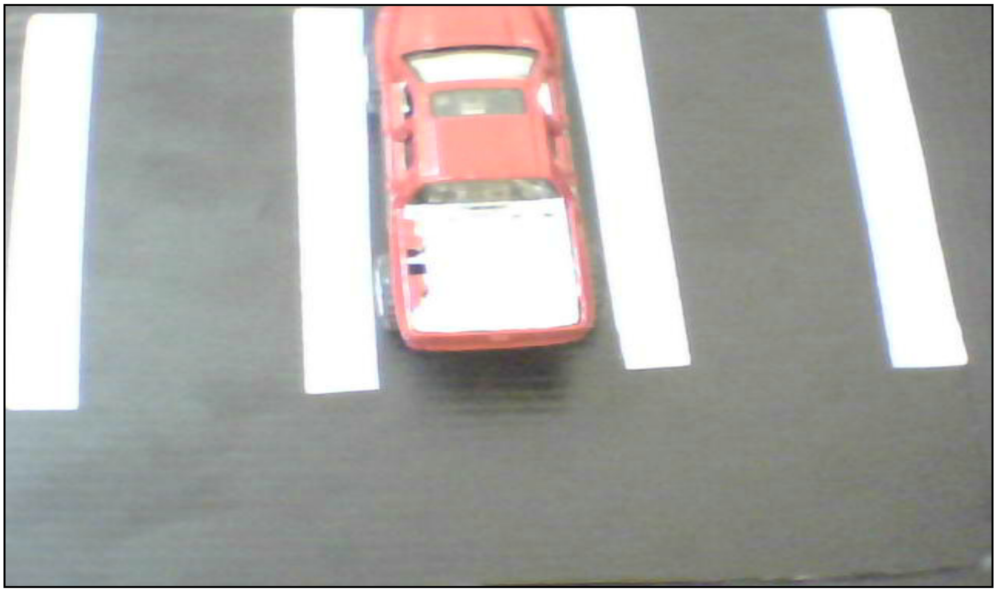


Figure 4.6(a) A real time image

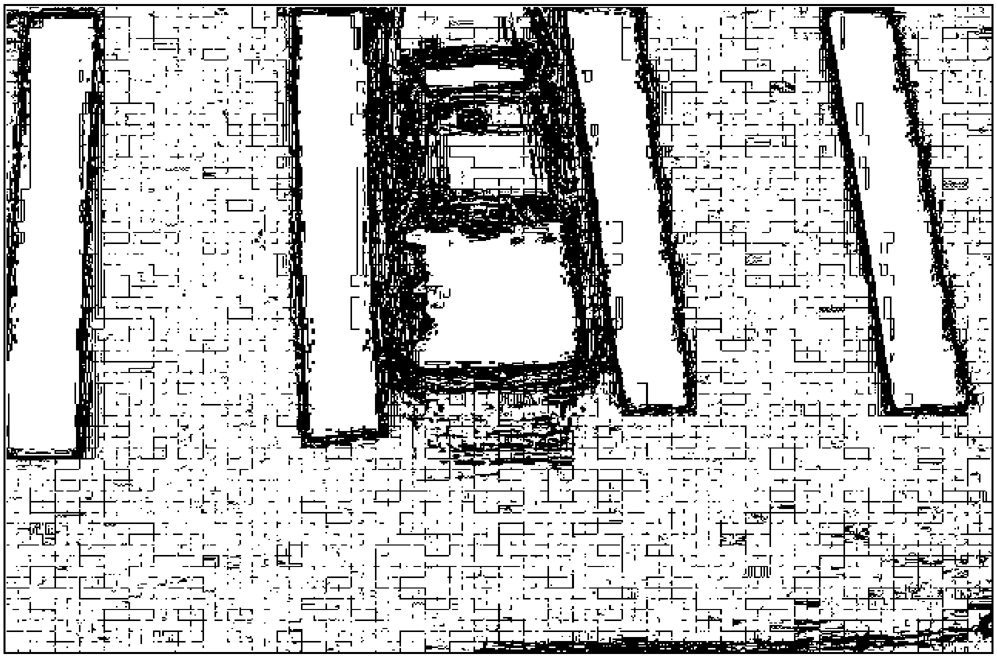


Figure 4.6(b) An edge detection image

Then, the system wills analyze the data on the second form. This form is shown a one car (Operator Image) is different with parking is empty (Source Image) and show result (Result Image) in figure 4.7. All figures below show different situation based on quantity vehicle on the parking park.

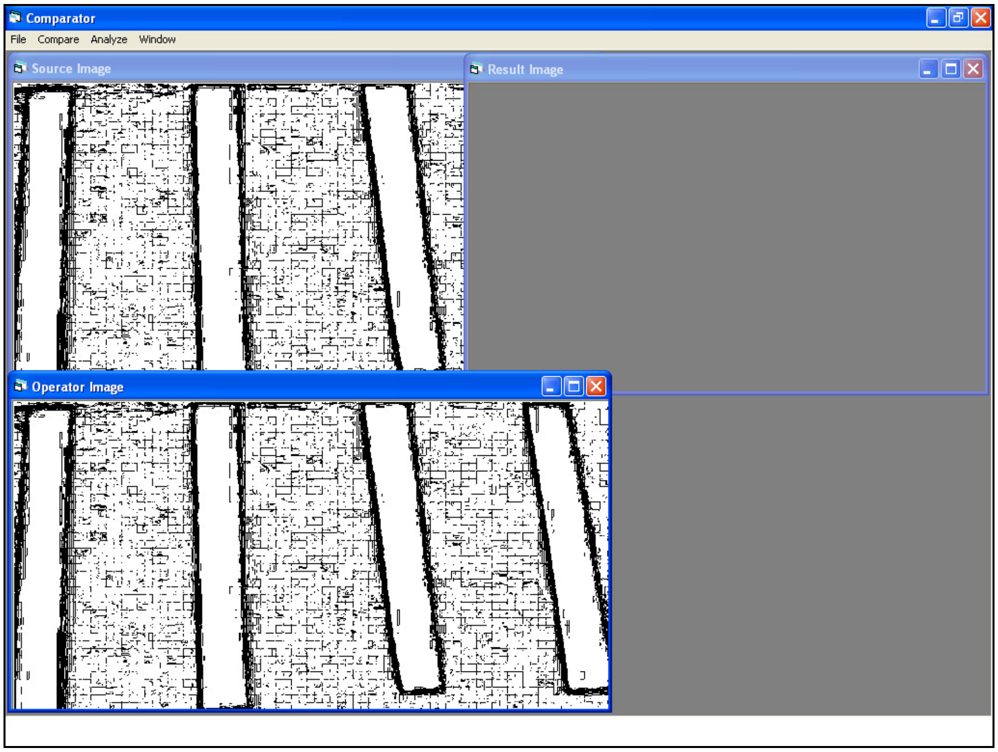


Figure 4.7 (a) No car analysis

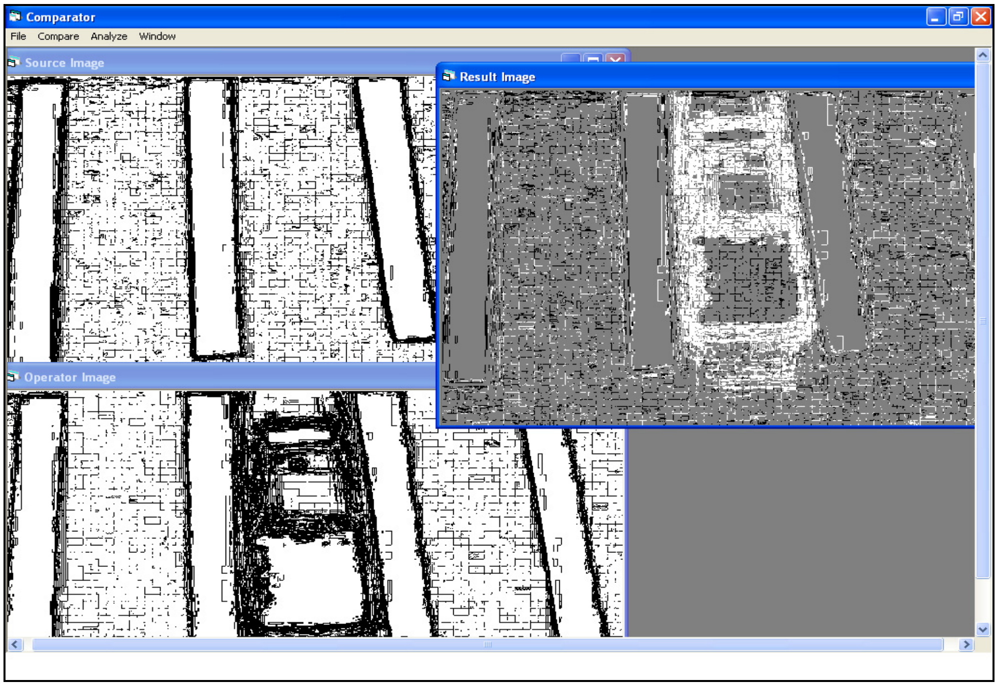


Figure 4.7 (b) One car analysis

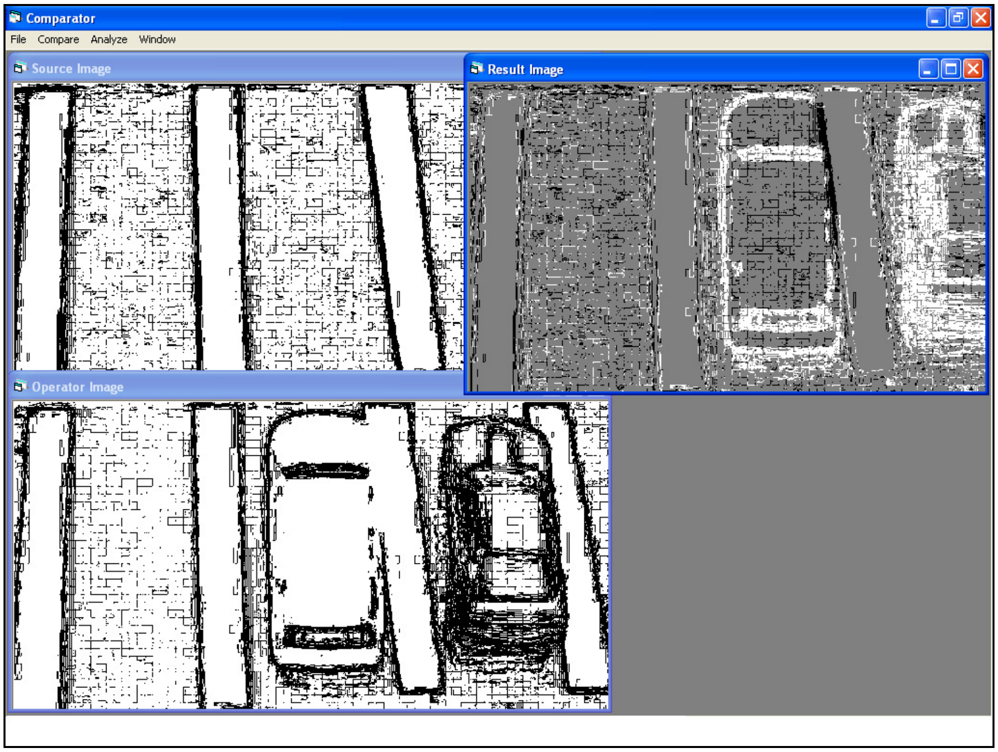


Figure 4.7 (c) Two car analysis

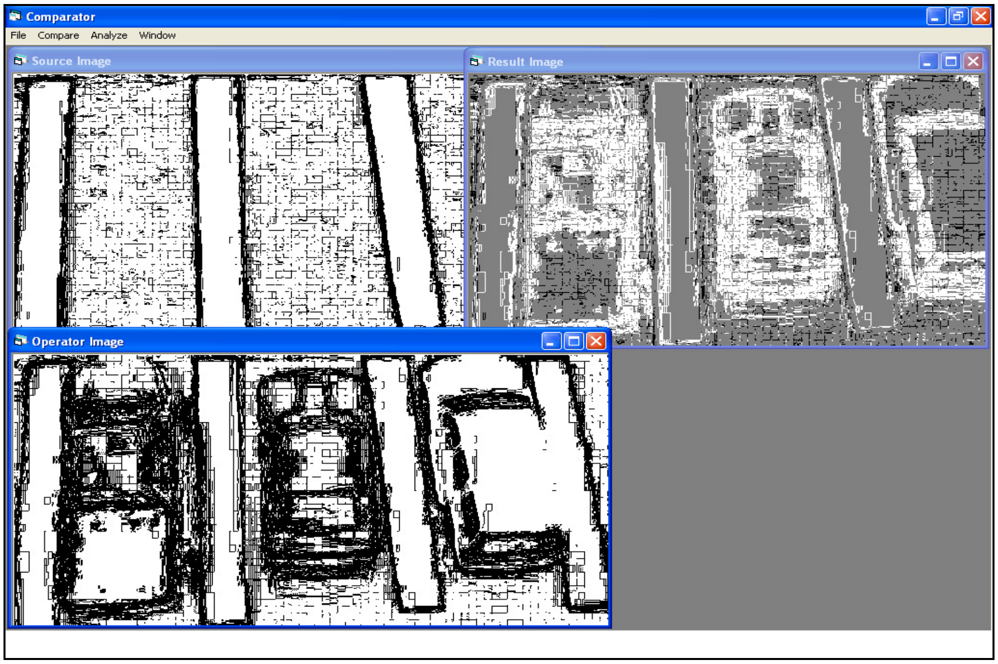
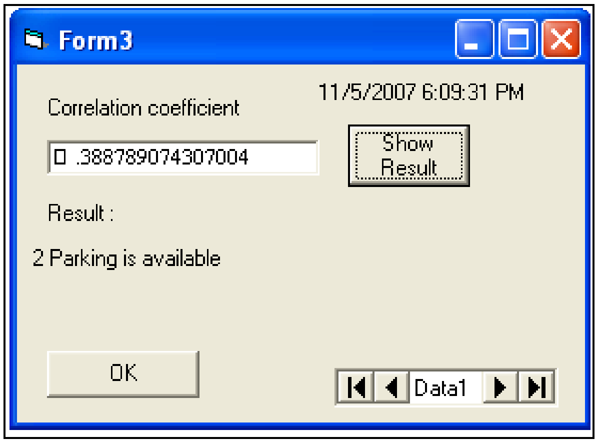


Figure 4.7 (d) Three car analysis

The next step it on Form 3 that will give a data to update database and inform to the client. This form based on analysis data at form 2 and calculates correlation coefficient on Result Image. Lastly, it will show the data and save it in database. After click "OK" button, the system will automatically back to initial program and do the same step. Figure 4.8 show the form 3.



11/5/2007 6:09:31 PM

Correlation coefficient: □ .388789074307004

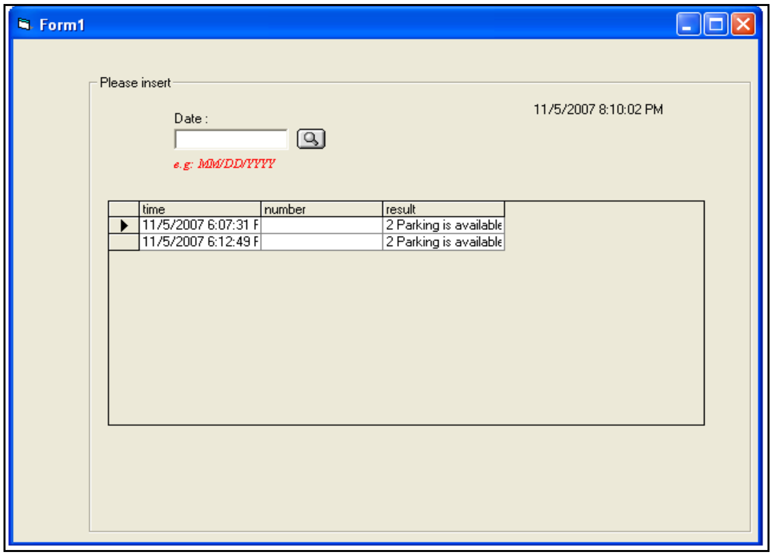
Result :

2 Parking is available

Figure 4.8 The Output (Data)

4.4 Client Page

User page is developing based on Visual Basic 6.0 coding that enable client to know availability of the parking park. This user page is not using any password or username to make it user friendly where all user outside can used this system. Figure 4.9 shows the main page of this system



Please insert

Date :

e.g: MM/DD/YYYY

11/5/2007 8:10:02 PM

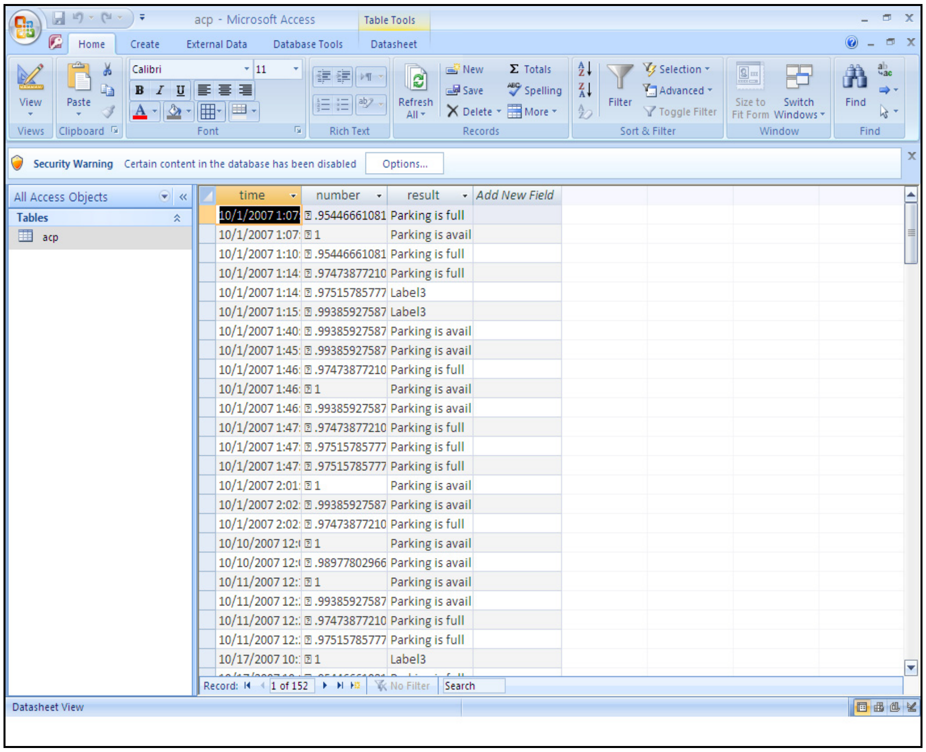
	time	number	result
▶	11/5/2007 6:07:31 F		2 Parking is available
	11/5/2007 6:12:49 F		2 Parking is available

Figure 4.9 The Client Form

4.5 Database

The system used the database to save the data from server and then sends back to client. The database will sends based on date that client want know availability parking. The database saved in table named “acp” and convert it in “database.mdb” that only can read on Visual Basic. The Figure 4.10 shows the database in Microsoft Access.

The database will integrate with visual basic by using Data Control. It is the object which links a form to a database and allows system to access the fields in the tables making up the database. It's called Data in the Toolbox**.** The database is connection information between administrator and client.



Certain content in the database has been disabled

time	number	result	Add New Field
10/1/2007 1:07	□ .95446661081	Parking is full	
10/1/2007 1:07:	□ 1	Parking is avail	
10/1/2007 1:10:	□ .95446661081	Parking is full	
10/1/2007 1:14:	□ .97473877210	Parking is full	
10/1/2007 1:14:	□ .97515785777	Label3	
10/1/2007 1:15:	□ .99385927587	Label3	
10/1/2007 1:40:	□ .99385927587	Parking is avail	
10/1/2007 1:45:	□ .99385927587	Parking is avail	
10/1/2007 1:46:	□ .97473877210	Parking is full	
10/1/2007 1:46:	□ 1	Parking is avail	
10/1/2007 1:46:	□ .99385927587	Parking is avail	
10/1/2007 1:47:	□ .97473877210	Parking is full	
10/1/2007 1:47:	□ .97515785777	Parking is full	
10/1/2007 1:47:	□ .97515785777	Parking is full	
10/1/2007 2:01:	□ 1	Parking is avail	
10/1/2007 2:02:	□ .99385927587	Parking is avail	
10/1/2007 2:02:	□ .97473877210	Parking is full	
10/10/2007 12:(	□ 1	Parking is avail	
10/10/2007 12:(	□ .98977802966	Parking is avail	
10/11/2007 12::	□ 1	Parking is avail	
10/11/2007 12::	□ .99385927587	Parking is avail	
10/11/2007 12::	□ .97473877210	Parking is full	
10/11/2007 12::	□ .97515785777	Parking is full	
10/17/2007 10::	□ 1	Label3	

Record: 1 of 152 No Filter Search

Figure 4.10 Database

4.6 Summary

As conclusion, this chapter shows all of the result from this project which is includes, Administrator Page, a Client Page and database.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

As a conclusion Automatic Vehicle Detection System developed in this project has gained result but still work in manually not 100% automatic. However, objectives in this project are archived. The system is running well and is capable of generating an optimize function.

The system will help to reduce a problem in parking management and improve parking industry. This system also based on server which controls the system and give output to client/users that access it.

This system can also convert a real image to edge detection and give an output either parking if full or empty. The system also can give how many parking spaces available based on analysis image by using correlation coefficient.

5.2 Recommendation for Future Work

Since the Automatic Vehicle Detection system is a new idea that uses image processing to analyze the parking situation, these are many improvements that can be done. Some recommendation for future includes:

1) Better result could be obtained by adding an image filtering algorithm. Color image could be used as an input since more information can be obtained for more filtering flexibility.

2) The system could also be modified to become fully automatic running'

3) Develop the system by be all in automatic system and users can get results by using WAP.

4) Since this project is a software part of the system, consideration to integrate with the hardware system must be taken into account.

From the above conclusion and recommendation, it is clear that there are the great possibilities in improving and also implementing the proposed system.

5.2.1 Costing and Commercialization

Generally, this project based on software. So, it does not involve any cost and all the software needed can be downloaded from the internet. To make it real, it can be implementing by upload to the server (web hosting). From the research that has been made, the renting fee for this service below RM50 per month.

This project has a potential to be commercialized especially to the parking company. The commercial value for this project is depends on developer himself, based on the programming level and customer requirements. As a conclusion, this project can be modified and commercialized by the customer to suite on any system.

.

REFERENCES

1. Qiu,Peihua, Image Processing and Jump Regression Analysis
ISBN 0-471-42099-9 (cloth) 2005

2. Pierre Francus, Image Analysis,Sediments and Paloeenvironment
ISBN 1-4020-2061-9 2004

3. Thomas M. Connolly, Carolyn E. Begg. – 3rd edition , Database systems: a practical approach to design, implementation, and management

4. Canny, J., A Computational Approach To Edge Detection, IEEE Trans. Pattern analysis and Machine Intelligence

5. Ian T. Young, Jan J. Gerbrands, Lucas J.Van Vliet Fundamentals of Image Processing, Paperback, ISBN 90-75691-01-7 (1995)

6. J. Maps (Jul 1993). "A computer-based data acquisition laboratory for undergraduates". American Journal of Physics 61 (7): 651-655.
doi:10.1119/1.17175.

7. Oliver Burkeman, Plug pulled on live website seen by millions" in The Guardian, 3rd January, 2004

8. Gonzalez, Rafael C. & Woods, Richard E. (2002). Thresholding. In Digital Image Processing, pp. 595–611. Pearson Education.
ISBN 81-7808-629-8

9. Michael Halvorson, Microsoft Visual Basic Professional 6.0 Step by Step ISBN 1-57231-809-0, QA76.73.B3H338 1998

10. Scott E Umbaugh, Computer Imaging: Digital Image Analysis and Processing, The CRC Press, 2005, CVIPtools CD-ROM with book, ISBN: 0-84-932919-1.

11. Charles D. Spencer, (1990). Digital Design for Computer Data Acquisition. Cambridge University Press. ISBN 0-521-37199-6

12. Russ J.C, 1999. The Image Processing Handbook.CRC Press, Boca Raton, Florida

13. Manifold System Release 5.50 (All Editions)- http://exchange.manifold.net/manifold/manuals /start_Threshold.htm

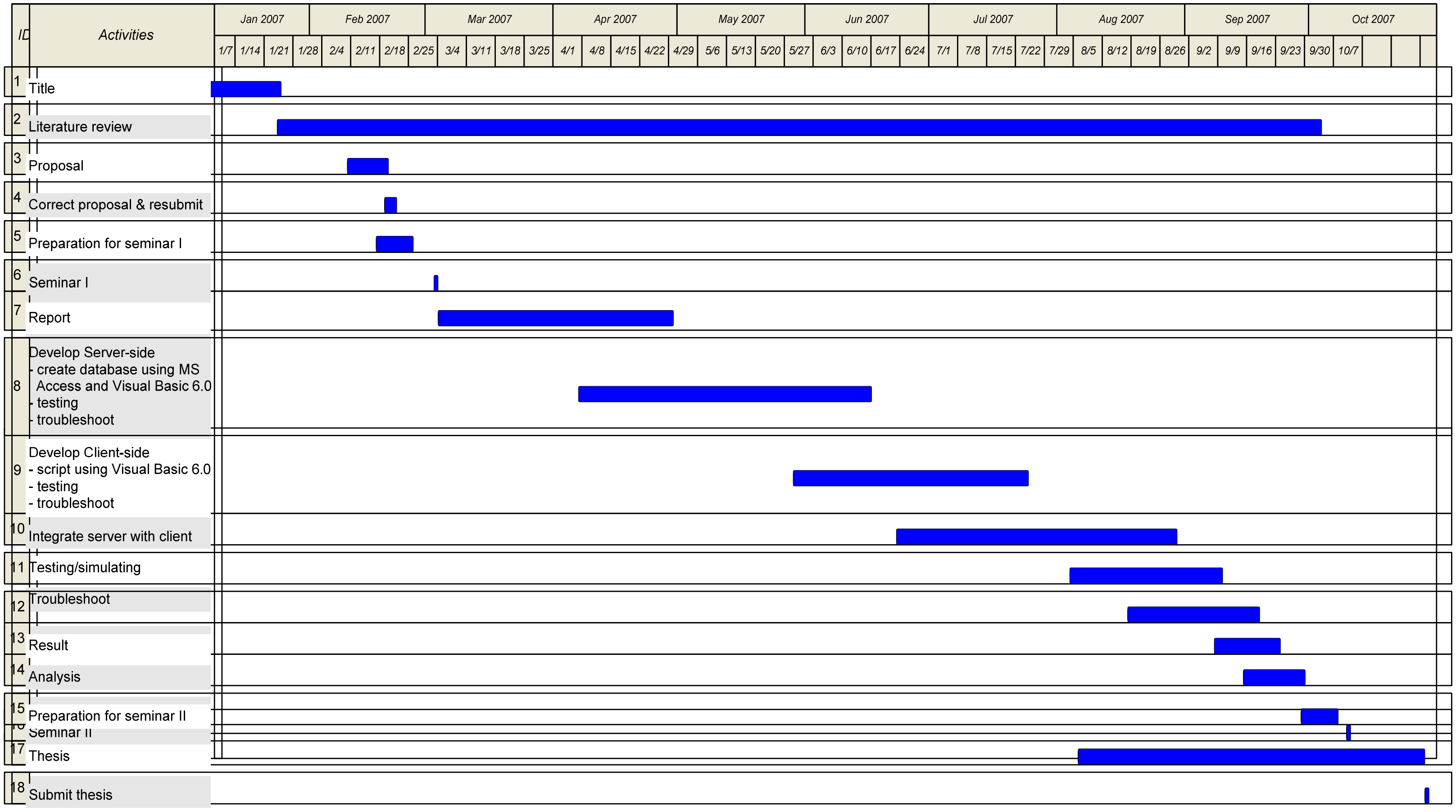
14. Dr.Dobb's Portal- http://www.ddj.com/about.html

15. Forum- http://www.codeproject.com/vbscript/VvvHardwareInfo.asp

APPENDIX A

GANTT CHART

APPENDIX A : PROJECT SCHEDULE



APPENDIX B

CODING SERVER

Coding Server Form

Form 1

```
Private Sub Command1_Click()
If Text1.Text = "" And Text2.Text = "" Then
MsgBox "Please insert username and password", , "Error"
Else
If Text1.Text = "admin" And Text2.Text = "12345" Then
Form1.Show
Unload Me
Else
msg = MsgBox("Invalid username or password , try again!", vbOKCancel)
If (msg = 1) Then
Text1.Text = ""
Text2.Text = ""
End If
End If
End If
End Sub
Private Sub Form_Load()
Label4.Caption = Now
Timer1.Interval = 1000
End Sub
Private Sub Timer1_Timer()
Label4.Caption = Now
End Sub
```

Form 2

```
Private Declare Function GetPixel Lib "gdi32" (ByVal hdc As Long, ByVal X As
Long, ByVal Y As Long) As Long
Private Declare Function SetPixel Lib "gdi32" (ByVal hdc As Long, ByVal X As
Long, ByVal Y As Long, ByVal crColor As Long) As Long
Private Declare Function capCreateCaptureWindow Lib "avicap32.dll" Alias
"capCreateCaptureWindowA" (ByVal lpszWindowName As String, ByVal dwStyle
As Long, ByVal X As Long, ByVal Y As Long, ByVal nWidth As Long, ByVal
nHeight As Long, ByVal hwndParent As Long, ByVal nID As Long) As Long
Dim Cancel As Boolean
Dim AveCol As Long
Dim Saving As Boolean
Private Sub Command1_Click()
Picture2.Picture = Picture1.Picture
End Sub
Private Sub Command2_Click()
STARTCAM

Progress.Show
Progress.start
Unload Progress
Command2.Enabled = False
Command3.Enabled = True


Picture1.AutoRedraw = True
Picture2.AutoRedraw = True
Timer1.Enabled = True
Timer2.Enabled = True
```

```
End Sub
Private Sub Command3_Click()
STOPCAM
ProgressBar1.value = 0
Command3.Enabled = False
Command2.Enabled = True
Picture1.Picture = LoadPicture("C:\Documents and Settings\Shahrul\Desktop\psm II
Final Project\nosignal.bmp")
Picture2.Picture = LoadPicture("C:\Documents and Settings\Shahrul\Desktop\psm II
Final Project\nosignal.bmp")
Label1.Caption = "0%"
End Sub
Private Sub Command4_Click()
List1.Clear
End Sub
Private Sub Command5_Click()
oprfrm.Show
Me.Hide
End Sub

Private Sub Form_Load()

InitDlgs
Dest.height = Source.height
Dest.width = Source.width
Picture1.width = 320 * Screen.TwipsPerPixelX
Picture1.height = 240 * Screen.TwipsPerPixelY
Picture2.width = 320 * Screen.TwipsPerPixelX
Picture2.height = 240 * Screen.TwipsPerPixelY
Picture1.Picture = LoadPicture("C:\Documents and Settings\Shahrul\Desktop\psm II
Final Project\nosignal.bmp")
Picture2.Picture = LoadPicture("C:\Documents and Settings\Shahrul\Desktop\psm II
Final Project\nosignal.bmp")
End Sub
```

```
Private Function Different(ByVal a As Long, ByVal b As Long) As Boolean

ar = a Mod 256: a = a \ 256
ag = a Mod 256: a = a \ 256
ab = a Mod 256: a = a \ 256

br = b Mod 256: b = b \ 256
bg = b Mod 256: b = b \ 256
bb = b Mod 256: b = b \ 256
sense = 255 - Slider1.value * 5

Different = (Sqr((ar - br) * (ar - br) + (ag - bg) * (ag - bg) + (ab - bb) * (ab - bb)) >
sense) 'formula for counting different
End Function
Private Sub Form_Unload(Cancel As Integer)
STOPCAM
SaveSetting "MotionDetect", "Param", "s1", Str(Slider1.value)
SaveSetting "MotionDetect", "Param", "s2", Str(Slider2.value)
End Sub
Private Sub StartEdgeDetect_Click()
End Sub
Private Sub Slider1_Change()
Slider1.value = "50"
End Sub
Private Sub Slider2_Change()
Slider2.value = "20"
End Sub
Private Sub Timer1_Timer()

SendMessage mCapHwnd, GET_FRAME, 0, 0
SendMessage mCapHwnd, COPY, 0, 0
Picture1.Picture = Clipboard.GetData: Clipboard.Clear
```

```
stepp = 3

Dim qan, qann As Long
qan = 0
qann = 0

For i = 1 To Picture1.width / Screen.TwipsPerPixelX Step stepp
For j = 1 To Picture1.height / Screen.TwipsPerPixelY Step stepp

If Different(Picture1.Point(i * stepp * Screen.TwipsPerPixelX, j * stepp *
Screen.TwipsPerPixelY), Picture2.Point(Screen.TwipsPerPixelX * i * stepp, j *
stepp * Screen.TwipsPerPixelY)) Then
Picture1.Circle (i * stepp * Screen.TwipsPerPixelX, Screen.TwipsPerPixelY * j *
stepp), 1, RGB(255, 0, 0)
qann = qann + 1
End If

Next
Next
Label1.Caption = Int(qann * 100 / 910) & "%"
ProgressBar1.value = Int(qann * 100 / 910)
End Sub

Sub STOPCAM()
DoEvents: SendMessage mCapHwnd, DISCONNECT, 0, 0
Timer1.Enabled = False
Timer2.Enabled = False
End Sub

Sub STARTCAM()

mCapHwnd = capCreateCaptureWindow("WebcamCapture", 0, 0, 0, 320, 240,
Me.hwnd, 0)
DoEvents
```

```
SendMessage mCapHwnd, CONNECT, 0, 0
SendMessage mCapHwnd, WM_CAP_DLG_VIDEOFORMAT, 0, 0
DoEvents
Slider1.value = GetSetting("MotionDetect", "Param", "s1", "0")
Slider2.value = GetSetting("MotionDetect", "Param", "s2", "0")

End Sub

Private Sub Timer2_Timer()
If ProgressBar1.value > 100 - Slider2.value * 2 Then
Beep
SavePicture Picture1.Picture, App.Path + "\Detected\" + Format(Date, "ddmmyyyy")
+ "__" + Format(time, "hhmmss") + ".bmp"
List1.AddItem "Saved in " + Str(time) + " " + Str(ProgressBar1.value) + "%   -->  " +
Format(Date, "ddmmyyyy") + "__" + Format(time, "hhmmss") + ".bmp"
End If



DoEvents
 Dest.Refresh
 Dim File As String '
 File = Open_File(Me.hwnd)
 If Trim(File) = "" Then Exit Sub
 Source.Picture = LoadPicture(File)
 Dest.height = Source.height
 Dest.width = Source.width
 Source.top = 0: Source.left = 0: Dest.top = 0: Dest.left = 0
 SScrollV.value = 0: SScrollH.value = 0: DScrollV.value = 0: DScrollH.value = 0

 If Source.height <= SourceContainer.height Then SScrollV.Enabled = False:
DScrollV.Enabled = False: GoTo Next1:
 SScrollV.Enabled = True
 SScrollV.max = Source.height - SourceContainer.height
 SScrollV.LargeChange = SScrollV.max / 5
 DScrollV.Enabled = True
 DScrollV.max = Source.height - SourceContainer.height
 DScrollV.LargeChange = DScrollV.max / 5
Next1:
 If Source.width <= SourceContainer.width Then SScrollH.Enabled = False: DScrollH.Enabled = False: GoTo Next2:
 SScrollH.Enabled = True
 SScrollH.max = Source.width - SourceContainer.width
 SScrollH.LargeChange = SScrollH.max / 5
 DScrollH.Enabled = True
 DScrollH.max = Source.width - SourceContainer.width
 DScrollH.LargeChange = DScrollH.max / 5
Next2:
Cancel = False
'StartEdgeDetect.Enabled = False
Dim Col As Long ' hold the colour of the pixel minus the colour of the pixel next to it
Dim Total As Long
Total = (Source.width / Screen.TwipsPerPixelX) * (Source.height / Screen.TwipsPerPixelY)
Dest.Cls
For X = 1 To Source.width \ Screen.TwipsPerPixelX 'loop through the x-pixels
 For Y = 1 To Source.height \ Screen.TwipsPerPixelY 'loop through the y-pixels
  Col = Abs(GetPixel(Source.hdc, X, Y) - GetPixel(Source.hdc, X, Y - 1)) ' hold the colour of the pixel minus the colour of the pixel on the top of it
  If Col > (Tolerance.value) ^ 3 Then Col = vbWhite Else Col = 0 ' choose if the colour is of high contrast
  If Invert.value = 0 Then Col = (vbWhite - Col) ' check for an invert
  If Col = 0 Then SetPixel Dest.hdc, X, Y, Col ' plot pixel
  Col = Abs(GetPixel(Source.hdc, X, Y) - GetPixel(Source.hdc, X - 1, Y)) ' hold the colour of the pixel minus the colour of the pixel on the left of it
  If Col > (Tolerance.value) ^ 3 Then Col = vbWhite Else Col = 0 ' choose if the colour is of high contrast
  If Invert.value = 0 Then Col = (vbWhite - Col) ' check for an invert
  If Col = 0 Then SetPixel Dest.hdc, X, Y, Col ' plot pixel
 Next Y 'loop through the y-pixels
 PercentDone.Caption = Int(((X * Y) / Total) * 100) & "%" 'calculate the percent
done.
 DoEvents
 Dest.Refresh
 If Cancel = True Then GoTo Finish:
Next X 'loop through the x-pixel
Next3:
'Dim File As String ' holds the file name
Beep
 'File = Save_File(Me.hwnd) 'show save dlg
 'If Trim(File) = "" Then MsgBox "File Not Saved, Invalid Filename.", vbCritical,
"Error": Exit Sub ' error in name
 'Saving = True ' start saving
 'Me.Caption = "Please Wait Saving....."
Dest.Picture = Dest.image 'set the picture to equal the image
SavePicture Dest.Picture, App.Path + "\process\" + Format(Date, "ddmmyyyy") +
"__" + Format(time, "hhmmss") + ".bmp"
Next4:
oprfrm.Show
Me.Hide

STOPCAM
ProgressBar1.value = 0
Command3.Enabled = False
Command2.Enabled = True

Picture1.Picture = LoadPicture("C:\Documents and Settings\Shahrul\Desktop\psm II
Final Project\nosignal.bmp")
Picture2.Picture = LoadPicture("C:\Documents and Settings\Shahrul\Desktop\psm II
Final Project\nosignal.bmp")
Label1.Caption = "0%"
```

```
Finish:
'StartEdgeDetect.Enabled = True
End Sub
Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)
 Do While Saving = True
  DoEvents
 Loop
 Unload Me
 End
End Sub
Private Sub Save_Click()
Dim File As String ' holds the file name
 File = Save_File(Me.hwnd) 'show save dlg
 If Trim(File) = "" Then MsgBox "File Not Saved, Invalid Filename.", vbCritical,
"Error": Exit Sub ' error in name
 Saving = True ' start saving
 Me.Caption = "Please Wait Saving....."
 Dest.Picture = Dest.image 'set the picture to equal the image
 SavePicture Dest.Picture, App.Path + "\process\" + Format(Date, "ddmmyyyy") +
"__" + Format(time, "hhmmss") + ".bmp"
 'Tmp.Caption = File '-- get rid of any unwanted chars (ie chr13, or 0)
 'File = Tmp.Caption '/
 'If LCase(right(File, 4) <> ".bmp") Then File = File & ".bmp" ' add the bmp on the
file
 'Call SavePicture(Dest(0).Picture, File) ' save the picture
 'Saving = False ' no longer saving
End Sub
Private Sub Load_Click()

End Sub
Sub CancelDraw_Click()
 Cancel = True
 'StartEdgeDetect.Enabled = True
End Sub
```

```
Private Sub SScrollH_Change()
 Source.left = -(SScrollH.value)
End Sub
Private Sub SScrollV_Change()
 Source.top = -(SScrollV.value)
End Sub
Private Sub DScrollH_Change()
 Dest.left = -(DScrollH.value)
End Sub
Private Sub DScrollV_Change()
 Dest.top = -(DScrollV.value)
End Sub

Private Function GreyFromColour(LongCol As Long) As Integer
 Dim Blue As Double, Green As Double, red As Double, GreenS As Double, BlueS
As Double, Num As Integer
 Blue = Fix((LongCol / 256) / 256)
 Green = Fix((LongCol - ((Blue * 256) * 256)) / 256)
 red = Fix(LongCol - ((Blue * 256) * 256) - (Green * 256))
 If uRed.value = 1 Then GreyFromColour = GreyFromColour + red: Num = Num +
1
 If uGreen.value = 1 Then GreyFromColour = GreyFromColour + Green: Num =
Num + 1
 If uBlue.value = 1 Then GreyFromColour = GreyFromColour + Blue: Num = Num
+ 1
 GreyFromColour = GreyFromColour / Num
End Function
Private Sub Tolerance_Change()
 Tolerance.value = "53"
End Sub
Private Sub Tolerance_Scroll()
 Tolerance.value = "53"
End Sub
Private Sub uGreen_Click()
```

```
 If uBlue.value = 0 _
 And uRed.value = 0 _
 And uGreen.value = 0 _
 Then uGreen.value = 1
End Sub
Private Sub uRed_Click()
 If uBlue.value = 0 _
 And uRed.value = 0 _
 And uGreen.value = 0 _
 Then uRed.value = 1
End Sub
Private Sub uBlue_Click()
 If uBlue.value = 0 _
 And uRed.value = 0 _
 And uGreen.value = 0 _
 Then uBlue.value = 1
End Sub
```

Form 3

```
Private Sub Command1_Click()
If Text1.Text = 1 Then
Label3.Caption = "3 Parking is available"
Else
If Text1.Text >= 0.31 And Text1.Text <= 0.36 Then
Label3.Caption = "1 Parking is available"
Else
If Text1.Text >= 0.365 And Text1.Text <= 0.399 Then
Label3.Caption = "2 Parking is available"
Else
If Text1.Text >= 0.244 And Text1.Text <= 0.3096 Then
```

```
Label3.Caption = "Parking is full"
End If
End If
End If
End If
End Sub

Private Sub Command2_Click()
Data1.Recordset.AddNew
Data1.Recordset.Fields("time").value = Label4.Caption
Data1.Recordset.Fields("number").value = Text1.Text
Data1.Recordset.Fields("result").value = Label3.Caption
Data1.Recordset.Update
Data1.Refresh
Form1.STARTCAM
Progress.Show
Progress.start
Unload Progress
Form1.Command2.Enabled = False
Form1.Command3.Enabled = True

Form1.Picture1.AutoRedraw = True
Form1.Picture2.AutoRedraw = True
Form1.Timer1.Enabled = True
Form1.Timer2.Enabled = True
Form1.Show
Unload Me

End Sub
Private Sub Form_Load()
Label4.Caption = Now
Timer1.Interval = 1000
Data1.DatabaseName = App.Path & "\Database1.mdb"
Data1.RecordSource = "select * from acp"
```

```
Data1.Refresh
End Sub
Private Sub Timer1_Timer()
Label4.Caption = Now
End Sub
```

Module for Webcam

```
Public Declare Function SendMessage Lib "user32" Alias "SendMessageA" (ByVal
hwnd As Long, ByVal wMsg As Long, ByVal wParam As Long, lParam As Any) As
Long
Public mCapHwnd As Long
Public Const CONNECT As Long = 1034
Public Const DISCONNECT As Long = 1035
Public Const GET_FRAME As Long = 1084
Public Const COPY As Long = 1054
Public Const WM_CAP_SET_VIDEOFORMAT = &H400 + 45
```

Module for Edge Detection

```
Public Declare Function GetOpenFileName Lib "comdlg32.dll" Alias
"GetOpenFileNameA" (pOpenfilename As OpenFileName) As Long
Public Declare Function GetSaveFileName Lib "comdlg32.dll" Alias
"GetSaveFileNameA" (pOpenfilename As OpenFileName) As Long

Public Type OpenFileName
  lStructSize As Long
  hwndOwner As Long
  hInstance As Long
  lpstrFilter As String
  lpstrCustomFilter As String
  nMaxCustFilter As Long
  nFilterIndex As Long
  lpstrFile As String
  nMaxFile As Long
  lpstrFileTitle As String
  nMaxFileTitle As Long
  lpstrInitialDir As String
  lpstrTitle As String
  flags As Long
  nFileOffset As Integer
  nFileExtension As Integer
  lpstrDefExt As String
  lCustData As Long
  lpfnHook As Long
  lpTemplateName As String
End Type
'Declare Variables
```

```
Public SaveFileDialog As OpenFileName
Public OpenFileDialog As OpenFileName
Private rv As Long
Private sv As Long
Public Function Open_File(hwnd As Long) As String
  rv& = GetOpenFileName(OpenFileDialog)
  If (rv&) Then
    Open_File = Trim$(OpenFileDialog.lpstrFile)
  Else
    Open_File = ""
  End If
End Function
Public Function Save_File(hwnd As Long) As String
  sv& = GetSaveFileName(SaveFileDialog)
  If (sv&) Then
    Save_File = Trim$(SaveFileDialog.lpstrFile)
  Else
    Save_File = ""
  End If
End Function
Private Sub InitSaveDlg()
 With SaveFileDialog
   .lStructSize = Len(SaveFileDialog)
   .hwndOwner = hwnd&
   .hInstance = App.hInstance
   .lpstrFilter = "Bmp Image File" + Chr$(0) + "*.Bmp"
   .lpstrFile = Space$(254)
   .nMaxFile = 255
   .lpstrFileTitle = Space$(254)
   .nMaxFileTitle = 255
   .lpstrInitialDir = App.Path
   .lpstrTitle = "Save Image..."
   .flags = 0
 End With
End Sub
Private Sub InitOpenDlg()
  With OpenFileDialog
   .lStructSize = Len(OpenFileDialog)
   .hwndOwner = hwnd&
   .hInstance = App.hInstance
   .lpstrFilter = "Image Files" + Chr$(0) + "*.bmp;*.jpg;*.pcx;*.gif"
   .lpstrFile = Space$(254)
   .nMaxFile = 255
   .lpstrFileTitle = Space$(254)
   .nMaxFileTitle = 255
   .lpstrInitialDir = App.Path
   .lpstrTitle = "Load Colour Image..."
   .flags = 0
  End With
End Sub
Public Sub InitDlgs()
 Call InitSaveDlg
 Call InitOpenDlg
End Sub
```

APPENDIX B

CODING CLIENT

Coding CLIENT form

```
Private Sub Combo1_Click()
'Label2.Caption = Text1.Text + " " + Text2.Text + " " + Combo1.Text
End Sub
Private Sub Command1_Click()
End Sub
Private Sub Form_Load()
Label1.Caption = Now
Timer1.Interval = 1000
Data1.DatabaseName = App.Path & "\Database1.mdb"
Data1.RecordSource = "SELECT * from acp "
Data1.Refresh
End Sub
Private Sub Image2_Click()
Data1.DatabaseName = "\\169.254.134.79\psm II Final Project\Database1.mdb"
If Text1.Text = "" Then
'Or Text2.Text = "" Or Combo1.Text = "" Then
MsgBox "please fill the blank", , "Error"
Else
Data1.DatabaseName = App.Path & "\Database1.mdb"
Data1.RecordSource = "Select * from acp where time LIKE '*" & Text1.Text & "*'"
Data1.Refresh
DBGrid1.Visible = True
'DataGrid1.Visible = True
'Image4.Visible = True
'Label2.Visible = True

Data1.Recordset.FindFirst " time LIKE '*" & Text1.Text & "*'"
If Data1.Recordset.NoMatch = False Then
```

```
 Text1.Text = Text
 'Command4.Enabled = True

  Else
  If Data1.Recordset.NoMatch = True Then
  MsgBox "No record for this time ", , "ACPSystem"
  Text1.Text = ""
  'Command4.Enabled = False
    End If
   End If
  End If

End Sub

Private Sub Text1_KeyPress(KeyAscii As Integer)
'Label2.Caption = Text1.Text + " " + Text2.Text + " " + Combo1.Text
End Sub

Private Sub Text2_KeyPress(KeyAscii As Integer)
'Label2.Caption = Text1.Text + " " + Text2.Text + " " + Combo1.Text
End Sub

Private Sub Timer1_Timer()
Label1.Caption = Now
End Sub
```