1-1-2007

Video based vehicle detection for advance warning Intelligent Transportation System

Naveen Chintalacheruvu
University of Nevada, Las Vegas

Follow this and additional works at: https://digitalscholarship.unlv.edu/rtds

Repository Citation
Chintalacheruvu, Naveen, "Video based vehicle detection for advance warning Intelligent Transportation System" (2007). UNLV Retrospective Theses & Dissertations. 2223.
https://digitalscholarship.unlv.edu/rtds/2223

This Thesis is brought to you for free and open access by Digital Scholarship@UNLV. It has been accepted for inclusion in UNLV Retrospective Theses & Dissertations by an authorized administrator of Digital Scholarship@UNLV. For more information, please contact digitalscholarship@unlv.edu.
VIDEO BASED VEHICLE DETECTION FOR ADVANCE WARNING
INTELLIGENT TRANSPORTATION SYSTEM

by

Naveen Chintalacheruvu

Bachelor of Engineering in Electronics and Instrumentation Engineering
VNR Vignana Jyothi Institute of Engineering and Technology
Jawaharlal Nehru Technological University, India
May 2004

A thesis submitted in partial fulfillment
of the requirements for the

Master of Science Degree in Electrical Engineering
Department of Electrical and Computer Engineering
Howard R. Hughes College of Engineering

Graduate College
University of Nevada, Las Vegas
December 2007
INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

UMI Microform 1452234
Copyright 2008 by ProQuest LLC.
All rights reserved. This microform edition is protected against unauthorized copying under Title 17, United States Code.
Thesis Approval
The Graduate College
University of Nevada, Las Vegas

December 4, 2007

The Thesis prepared by

Naveen Chintalacheruvu

Entitled

"Video Based Vehicle Detection for Advance Warning
Intelligent Transportation System"

is approved in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering

M. Vani
Examination Committee Chair

Dean of the Graduate College

Examination Committee Member

Examination Committee Member

Graduate College Faculty Representative

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
ABSTRACT

Video Based Vehicle Detection for Advance Warning Intelligent Transportation System

by

Naveen Chintalacheruvu

Dr. Venkatesan Muthukumar, Examination Committee Chair
Associate Professor of Electrical and Computer Engineering,
University of Nevada, Las Vegas

Video based vehicle detection and surveillance technologies are an integral part of Intelligent Transportation System (ITS), due to its non-intrusiveness and capability of capturing global and specific vehicle behavior data. The initial goal of this thesis is to develop an efficient advance warning ITS system for detection of congestion at work zones and special events based on video detection. The goals accomplished by this thesis are: 1) successfully developed the advance warning ITS system using off-the-shelf components and 2) Develop and evaluate an improved vehicle detection and tracking algorithm. The advance warning ITS system developed includes many off-the-shelf equipments like “Autoscope” (video based vehicle detector), Digital Video Recorders, RF Transceivers, high gain Yagi antennas, variable message signs and interface processors. The video based detection system used requires calibration and fine tuning of configuration parameters for accurate results. Therefore, an in-house video based vehicle detection system was developed using the Corner Harris algorithm to eliminate the need of complex calibration and contrasts modifications. The algorithm was

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
implemented using OpenCV library on a Arcom’s Olympus Windows XP Embedded development kit running WinXPE operating system. The algorithm performance is for accuracy in vehicle speed and count is evaluated. The performance of the proposed algorithm is equivalent or better to the Autoscope system without any modifications to calibration and lamination adjustments.
TABLE OF CONTENTS

ABSTRACT ..................................................................................................................................... iii

LIST OF FIGURES ........................................................................................................................ vii

ACKNOWLEDGEMENTS ........................................................................................................... ix

CHAPTER 1 INTRODUCTION ................................................................................................. 1
  1.1 Intelligent Transportation Systems .............................................................................. 1
  1.2 Vehicle Detection Systems .......................................................................................... 2
  1.3 Issues in current Vehicle Detection Systems ............................................................. 2
  1.4 Autoscope and its Drawbacks ...................................................................................... 3
  1.5 Vehicle Detection using Harris Corner Method (HCM) ......................................... 5
  1.6 Implementation of Harris Corner Method ................................................................. 6
  1.7 Thesis Organization ....................................................................................................... 6

CHAPTER 2 VEHICLE DETECTION AND TRACKING ......................................................... 7
  2.1 Introduction ...................................................................................................................... 7
  2.2 Classification of Vehicle Detection .............................................................................. 7
    2.2.1 Intrusive Sensors ................................................................................................... 8
    2.2.2 Non-Intrusive Sensors ........................................................................................ 12
    2.2.3 Video Image Processing Approaches .................................................................. 18
  2.3 Vehicle Tracking ........................................................................................................... 33
    2.3.1 Methodologies in Object Tracking .................................................................. 34
  2.4 Summary ......................................................................................................................... 41

CHAPTER 3 SYSTEM SETUP AND EQUIPMENT .................................................................. 42
  3.1 Components of the QD System and AWI System .................................................. 42
    3.1.1 Camera Unit .......................................................................................................... 44
    3.1.2 Digital Video Recorder ....................................................................................... 44
    3.1.3 Battery Power Supply Unit ................................................................................ 45
    3.1.4 RF Transceiver Unit ........................................................................................... 46
    3.1.5 Communication between Stations ................................................................... 47
    3.1.6 Autoscope Interface Processor Unit ............................................................... 49
    3.1.7 Autoscope Rackvision ........................................................................................ 50
    3.1.8 Autoscope ............................................................................................................. 52
    3.1.9 VMS Interface Processor ................................................................................... 70
    3.1.10 Improved VMS Interface Processing .............................................................. 72
  3.2 Harris Corner based Vehicle Detection System ...................................................... 73
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1.1</td>
<td>Vehicle Detection Approach in Autoscope</td>
<td>4</td>
</tr>
<tr>
<td>Figure 1.2</td>
<td>Harris Corner Algorithm</td>
<td>5</td>
</tr>
<tr>
<td>Figure 2.1</td>
<td>Road Tube Configurations for Single Lane Highways</td>
<td>9</td>
</tr>
<tr>
<td>Figure 2.2</td>
<td>Loop Detector</td>
<td>10</td>
</tr>
<tr>
<td>Figure 2.3</td>
<td>Piezoelectric Sensor</td>
<td>11</td>
</tr>
<tr>
<td>Figure 2.4</td>
<td>Perturbation of Earth's Magnetic Field by a Ferrous Metal Vehicle</td>
<td>12</td>
</tr>
<tr>
<td>Figure 2.5</td>
<td>Microwave Radar</td>
<td>13</td>
</tr>
<tr>
<td>Figure 2.6</td>
<td>Infrared sensor geometry</td>
<td>14</td>
</tr>
<tr>
<td>Figure 2.7</td>
<td>TC-30C ultrasonic range-measuring sensor</td>
<td>15</td>
</tr>
<tr>
<td>Figure 2.8</td>
<td>Passive Acoustic Array Sensor</td>
<td>16</td>
</tr>
<tr>
<td>Figure 2.9</td>
<td>Autoscope - Video Image Processor</td>
<td>17</td>
</tr>
<tr>
<td>Figure 2.10</td>
<td>Two orientations of a object</td>
<td>35</td>
</tr>
<tr>
<td>Figure 2.11</td>
<td>3-D Plot</td>
<td>36</td>
</tr>
<tr>
<td>Figure 2.12</td>
<td>Example of Robust Representation</td>
<td>37</td>
</tr>
<tr>
<td>Figure 2.13</td>
<td>Chromatic representation</td>
<td>37</td>
</tr>
<tr>
<td>Figure 2.14</td>
<td>Object Histograms</td>
<td>39</td>
</tr>
<tr>
<td>Figure 3.1</td>
<td>Block Diagram of the System</td>
<td>43</td>
</tr>
<tr>
<td>Figure 3.2</td>
<td>Snap shot of the components used in the system</td>
<td>43</td>
</tr>
<tr>
<td>Figure 3.3</td>
<td>Two 12V deep-cycle batteries connected in parallel and serial</td>
<td>46</td>
</tr>
<tr>
<td>Figure 3.4</td>
<td>4 Element Yagi Antenna</td>
<td>47</td>
</tr>
<tr>
<td>Figure 3.5</td>
<td>VMS Interface Processor Algorithm</td>
<td>48</td>
</tr>
<tr>
<td>Figure 3.6</td>
<td>Autoscope Interface Processor Algorithm</td>
<td>49</td>
</tr>
<tr>
<td>Figure 3.7</td>
<td>a) Basic Stamp Module and b) Carrier Board with Serial interface</td>
<td>50</td>
</tr>
<tr>
<td>Figure 3.8</td>
<td>Calibration of autoscope</td>
<td>54</td>
</tr>
<tr>
<td>Figure 3.9</td>
<td>Presence Detectors</td>
<td>56</td>
</tr>
<tr>
<td>Figure 3.10</td>
<td>Presence Detector Configuration</td>
<td>56</td>
</tr>
<tr>
<td>Figure 3.11</td>
<td>Speed Detector</td>
<td>59</td>
</tr>
<tr>
<td>Figure 3.12</td>
<td>Speed Detector Configuration</td>
<td>59</td>
</tr>
<tr>
<td>Figure 3.13</td>
<td>Count Detector</td>
<td>62</td>
</tr>
<tr>
<td>Figure 3.14</td>
<td>Count Detector Configuration</td>
<td>63</td>
</tr>
<tr>
<td>Figure 3.15</td>
<td>Boolean Detector Function</td>
<td>66</td>
</tr>
<tr>
<td>Figure 3.16</td>
<td>Boolean Detector Function block diagram</td>
<td>66</td>
</tr>
<tr>
<td>Figure 3.17</td>
<td>Autoscope Sub-system block diagram</td>
<td>69</td>
</tr>
<tr>
<td>Figure 3.18</td>
<td>Autoscope Sub-system operation Flow</td>
<td>70</td>
</tr>
<tr>
<td>Figure 3.19</td>
<td>VMS Sub-system block diagram</td>
<td>71</td>
</tr>
<tr>
<td>Figure 3.20</td>
<td>VMS Sub-system operation Flow</td>
<td>71</td>
</tr>
<tr>
<td>Figure 3.21</td>
<td>Improved VMS sub-system block diagram</td>
<td>72</td>
</tr>
<tr>
<td>Figure 3.22</td>
<td>Snapshot of VB Application Replacing VMS Interface Processor</td>
<td>72</td>
</tr>
</tbody>
</table>
ACKNOWLEDGMENTS

This thesis is dedicated to my parents and siblings, together who have crafted my life so beautifully. I am solely indebted to Dr. Venkatesan Muthukumar, my guru, for trusting my abilities and considering me for mentorship. His interaction during projects and field trips has significantly influenced my professional as well as personal life in more ways than one.

I express my gratitude to Dr. Hualiang Teng for his financial support. My research work has been bolstered by Dr. Emma Regentova with her conceptual ideas which served as backbone to my thesis and a big Thanks to her. Special thanks to Ajay Kumar Mandava for his image processing concepts’ support, Kunal Metkar for Visual C++ guidance, Sean Jahanpour for his timely and trivial guidance in Visual Basic coding.

I thank god for gifting me a sister, who has been my virtual Google on C++ programming and contributed towards my documentation. Positive attitudes such as Ashwini Raina, Shruti Patil, and Gopinath Balakrishnan have been very contagious and inspiring. I am also thankful to Shankar Neelakrishnan, Dawid Zydek, Arun Reddy Toomu, Ramakrishna Gopalakrishnan, Nachiket Jugade, Hongyan Wang and Nirup for making up B348 lab, a Temple of learning.
CHAPTER 1

INTRODUCTION

1.1 Intelligent Transportation Systems

The Surface Transportation System of the United States has estimated a travel demand increase of 30% over the next ten years [Proper & Cheslow, 1997]. In order to prevent congestion at current levels from getting worse, one of the options is to develop alternatives that increase capacity by improving the efficiency of the existing transportation system. This option focuses on building fewer lane-miles, while investing in Intelligent Transportation Systems (ITS) infrastructure. In 1991, the Intermodal Surface Transportation Efficiency Act resulted in the formation of the Federal Intelligent Transportation Systems (ITS) program to address ways to deal with the increase in travel demand on the nation’s transportation systems using the second option.

The goals of ITS are:

a) Enhance public safety.

b) Reduce congestion.

c) Improved access to travel and transit information.

d) Generate cost savings to motor carriers, transit operators, toll authorities.

e) Reduce detrimental environmental impacts

ITS include the following technologies: sensor, communication, and traffic control.

These technologies assist states, cities, and towns nationwide to meet increasing demands
on the surface transportation system. Vehicle detection and surveillance technologies are an integral part of ITS, since they gather all or part of the data that are used in ITS.

1.2 Vehicle Detection Systems

Vehicle Detection System may be defined as a system which is capable of detecting vehicles and measure traffic parameters namely count, speed etc. Vehicle detection by video cameras is one of the most promising new technologies for wireless large-scale data collection and implementation of advanced traffic control and management schemes such as vehicle guidance/navigation. Vehicle detection is a basis for vehicle tracking. The correct vehicle detection results in better tracking.

1.3 Issues in Current Vehicle Detection Systems

Some of the problem areas encountered while employing off-the-shelf vehicle detectors are:

a) Excessive vehicle detector system hardware cost.

b) Excessive vehicle detector system installation cost.

c) Vehicle detector accuracy and precision are insufficient to meet the requirements of more advanced control systems.

d) Unavailability of environmental, ambient conditions and external noise levels to guide vehicle detector design and evaluation.

e) Off-the-shelf vehicle detectors employed in present traffic control systems are magnetic detectors, inductive loop detectors, continuous and pulsed sonic detectors and Doppler radar detectors.
Modern computer controlled traffic systems have more complex vehicle detection requirements than those for normal traffic-actuated controllers for traffic signals, for which many off-the-shelf vehicle detectors were designed. Some of the traffic parameters required are presence, passage (function of vehicle length), speed, acceleration, headway, average speed, vehicle length.

Many commercial off-the-shelf video based vehicle detectors are available. Autoscope and Iteris are most commonly used vehicle detectors used in U.S.

1.4 Autoscope and its Drawbacks

Autoscope, a video based vehicle detection system detects traffic in many locations (i.e., multiple spots) within the camera's field-of-view. Vehicle Detection Algorithm Approach employed in Autoscope is shown in Figure 1.1.

In the first step, an estimate of the background detector signature is accrued from a finite impulse response (FIR) filter. In the background suppression block, the current Background estimate is subtracted from the incoming detector data. Once the detector data have been reduced to a selected set of features, the next level of processing transforms the set of feature measurements \( F_1, F_2, \ldots, F_n \) to a corresponding set of instantaneous logic states \( S_1, S_2, \ldots, S_n \). These logic states indicate the presence or absence of important conditions in the detector footprint. In the final stage, vehicle presence processing consists of a tracker module which coalesces the time series of decision states into a high confidence presence signal. The demerits of Autoscope are shown below:
a) Vulnerability to viewing obstructions; inclement weather; shadows; vehicle projection into adjacent lanes; occlusion; day to-night transition; vehicle/road contrast.

b) Susceptible to camera motion caused by strong winds.

c) Average speed calculated by Autoscope is (approx.) 5 miles less than Radar gun average speed for our video recorded on I-15 during Feb 2007.

Figure 1.1: Vehicle Detection Algorithm Approach in Autoscope

1.5 Vehicle Detection using Harris Corner Method (HCM).

To suppress the above stated problems, corners of the moving objects are considered as a basis for vehicle detection. Harris Corner detection algorithm is chosen to fulfill the
requirement, since it has advantages like robust to noise, clutter, invariance to image transformations and robust to illumination changes. This algorithm identifies object corners present within the image using the following procedure.

a) The technique first identifies vertical and horizontal edges using a Sobel type edge detector.

b) These edges are then blurred to reduce the effect of any image noise.

c) The resulting edges are then combined together to form an energy map that contains peaks and valleys.

d) The peaks indicate the presence of a corner.

The algorithm flow is shown in Figure 1.2. The result of object detection forms a basis for our vehicle detection and tracking system.

![Figure 1.2: Harris Corner algorithm](image)

1.6 Implementation of Harris Corner Method

The Harris Corner Method is implemented using Microsoft Visual C++ (also known as MSVC) toolset using OpenCV image processing library. Vehicle detection and
tracking system was developed as a standalone embedded system on Arcom’s Development kit. MSVC is a commercial integrated development environment (IDE) product engineered by Microsoft for the C and C++ programming languages. It has tools for developing and debugging C++ code, especially that is written for the Microsoft Windows API, the DirectX API, and the Microsoft .NET Framework.

OpenCV means Intel Open Source Computer Vision Library. It is a collection of C functions and a few C++ classes that implement some popular Image Processing and Computer Vision algorithms.

1.7 Thesis Organization

The remainder of this thesis is organized as follows: In Chapter 2, Vehicle Detection and tracking, methods and approaches adopted earlier are introduced. Chapter 3 discusses the vehicle detection setup and equipment. Harris Corner Detection algorithm and its application is explained in Chapter 4. Evaluation of the proposed system is given in Chapter 5. Chapter 6 contains a summary of the important results of this thesis and finally Chapter 7 ends up with a conclusion and future work of this thesis.
CHAPTER 2

VEHICLE DETECTION AND TRACKING

2.1 Introduction

Motion detection and tracking are two tasks that play a fundamental role in video surveillance systems, transportation systems, military applications, gaming systems, etc. In this chapter, the basics of vehicle tracking are discussed. First classification of earlier vehicle detection methods is discussed. Delving further, vehicle tracking classification methods are discussed.

"Vehicle Detection" is a process of detecting the presence or absence of a vehicle in the video sequence. "Vehicle Tracking" is defined as finding the location of a vehicle of the scene on each frame of the video sequence. The result of detection is used as initialization process for tracking.

2.2 Classification of Vehicle Detection

Vehicle detection and tracking technologies can be divided into three major components:

a) The transducer.

b) A signal processing device.

c) A data processing device.

The transducer detects the passage or presence of a vehicle or its axles. The signal-processing device typically converts the transducer output into an electrical signal. The
data-processing device usually consists of computer hardware and firmware that converts the electrical signal into traffic parameters. Typical traffic parameters include vehicle presence, count, speed, vehicle classification, inter vehicle distance, headway, occupancy, weight and travel time.

Vehicle detection techniques are broadly classified into two categories:

a) Intrusive and
b) Non-Intrusive sensors.

2.2.1 Intrusive Sensors

Intrusive sensors include inductive loops, magnetometers, micro-loop probes, pneumatic road tubes, piezoelectric cables and other weigh-in-motion sensors. These devices are installed directly on the pavement surface, in saw-cuts or holes in the road surface by tunneling under the surface or by anchoring directly to the pavement surface as is the case of pneumatic road tubes.

The drawbacks of intrusive sensors include disruption of traffic during installation and repair and failures associated with installations on poor road surfaces and use of sub-standard installation procedures. Some of intrusive sensors for vehicle detection include:

a) Pneumatic Road Tube
b) Inductive loop Detector
c) Piezoelectric sensor
d) Magnetic sensor

a) Pneumatic Road Tube sensor

Pneumatic road tube sensor sends a burst of air pressure along a rubber tube when a vehicle's tires pass over the tube. The pulse of air pressure closes an air switch, producing
an electrical signal that is transmitted to a counter or analysis software. The pneumatic road tube sensor is portable, using lead-acid or gel or other rechargeable batteries as a power source. Figure 2.1 shows two road tube configurations for vehicle detection and classification.

![Figure 2.1: Road tube configurations for single lane highways.](Photograph courtesy of Time Mark, Inc., Salem OR)

Advantages

i) Quick installation, low power utilization.

ii) Low cost and simple maintenance.

Disadvantages

i) Inaccurate axle counting when truck and bus volumes are high.

ii) Temperature sensitivity of the air switch.

iii) Cut tubes resulting from vandalism and wear and tear produced by truck tires.

b) Inductive Loop Detectors

The inductive loop detector (ILD) is the most common sensor used in traffic management applications. Its size and shape vary, including the 5-ft by 5-ft or 6-ft by 6-ft square loops, 6-ft diameter round loops, and rectangular configurations having a 6-ft width and variable length (Figure 2.2).
The wire loop is excited with signals whose frequencies range from 10 KHz to 50 KHz and functions as an inductive element in conjunction with the electronic unit. When a vehicle stops on or passes over the loop, the inductance of the loop is decreased. The decreased inductance increases the oscillation frequency and causes the electronics unit to send a pulse to the controller, indicating the presence or passage of a vehicle.

Advantages

i) Flexible design.

ii) The equipment cost of inductive loop sensors is low.

Disadvantages

i) Expensive and inconvenient to install.

ii) Degrades the condition of pavements and roads.

iii) Unreliable over the time and no visual surveillance capabilities.

c) Piezoelectric Sensors

A piezoelectric is a specially processed material capable of converting kinetic energy to electrical energy. Some polymer materials exhibit these properties and are ideal in the construction of piezoelectric sensors (Figure 2.3).
Piezoelectric materials generate an electrical signal (voltage) when subjected to mechanical impact or vibration. Electrical charges of opposite polarity appear at the parallel faces and induce a voltage. The measured voltage is proportional to the force or weight of the vehicle. The magnitude of the piezoelectric effect depends upon the direction of the force in relation to the axes of the crystal.

Figure 2.3: Piezoelectric sensor
(Drawing courtesy of IRD, Inc., Saskatoon, SK)

Advantages

i) Differentiates individual vehicles with extreme precision.

ii) Determines the classification and weight of the vehicle.

Disadvantages

i) Disruption of traffic for installation.

ii) Sensitive to pavement temperature and vehicle speed.

d) Magnetic Sensors

Magnetic sensors are passive devices that indicate the presence of a metallic object by detecting the perturbation (known as a magnetic anomaly) in the Earth’s magnetic field created by the object (Figure 2.4).
Advantages

i) Fluxgate magnetometer is less susceptible than loops to stresses of traffic.

ii) The induction magnetometer can be used where loops are not feasible.

Disadvantages

i) Magnetic detectors cannot generally detect stopped vehicles.

ii) Some models have small detection zones.

2.2.2 Non-Intrusive Technologies

Non-intrusive technologies are those that do not require the installation of the sensor directly onto or into the road surface. The sensors for non-intrusive technologies are mounted overhead or on the side of the road.

a) Microwave Radar

Roadside-mounted microwave radar transmits energy toward an area of the roadway from an overhead antenna. The beam width or area in which the radar energy is transmitted is controlled by the size and the distribution of energy across the aperture of the antenna. The manufacturer usually establishes the design constraints. When a vehicle passes through the radar aperture, a portion of the transmitted energy is reflected back
towards the antenna. The energy then enters a receiver where the detection is made and vehicle data, such as volume, speed, occupancy, and length are calculated (Figure 2.5).

![Microwave Radar Diagram](image)

**Figure 2.5: Microwave radar**  
(Courtesy: The Vehicle Detector Clearinghouse, New Mexico SU)

Advantages

i) Insensitive to inclement weather.

ii) Microwave radar provides a direct measurement of speed. Also, multiple lane operation models are available.

Disadvantages

i) Microwave radar applications must insure that antenna beamwidth and transmitted waveform are suitable for the application.

ii) Doppler microwave sensors have been found to perform poorly at intersection locations as volume counters.

b) Infrared Sensors

The sensors are mounted overhead to view approaching or departing traffic or traffic from a side-looking configuration. Infrared sensors (shown in Figure 2.6) are used for signal control; volume, speed, and classification measurement, as well as detecting pedestrians in crosswalks.
Advantages

i) Some advantages of active infrared sensors are that they transmit multiple beams for accurate measurement of vehicle position, speed and classification.

ii) Multi-zone passive infrared sensors measure speed.

Disadvantages

i) Atmospheric particulates and inclement weather can scatter or absorb energy that would otherwise reach the focal plane

ii) Performance degradation in rain and snow.

c) Ultrasonic Sensors

Ultrasonic sensors transmit pressure waves of sound energy at a frequency between 25 and 50 KHz, which are above the human audible range. Most ultrasonic sensors, such as the model shown in Figure 2.7, operate with pulse waveforms and provide vehicle count, presence, and occupancy information. Pulse waveforms measure distances to the road surface and vehicle surface by detecting the portion of the transmitted energy that is reflected towards the sensor from an area defined by the transmitter's beamwidth. When
a distance other than background road surface is measured, the sensor detects the presence of a vehicle.

Figure 2.7: TC-30C ultrasonic range-measuring sensor. (Courtesy of Microwave Sensors, Ann Arbor, MI)

Advantages

i) Installation of ultrasonic sensors does not require an invasive pavement procedure.

ii) Some models feature multiple lane operation.

Disadvantages

i) Temperature change and extreme air turbulence may affect the performance of ultrasonic sensors.

ii) Large pulse repetition periods may degrade occupancy measurement on freeways with vehicles traveling at moderate to high speeds.

d) Passive Acoustic Array Sensors

Acoustic sensors (shown in Figure 2.8) measure vehicle passage, presence, and speed by detecting acoustic energy or audible sounds produced by vehicular traffic from a variety of sources within each vehicle and from the interaction of a vehicle tires with the
road. When a vehicle passes through the detection zone, an increase in sound energy is recognized by the signal-processing algorithm and a vehicle presence is detected. When the vehicle leaves the detection zone, the sound energy level drops below the detection threshold and the vehicle presence signal is terminated. Sounds from locations outside the detection zone are attenuated.

Figure 2.8: Passive Acoustic Array Sensor

Advantages

i) Acoustic sensors are insensitive to precipitation.

ii) Multiple lane operation is available in some models.

Disadvantages

i) Cold temperatures affect the accuracy of the data from acoustic sensors.

ii) Specific models are not recommended with slow moving vehicles in stop and go traffic.

c) Video Image Processor

Video cameras were introduced to traffic management for roadway surveillance because of their ability to transmit closed circuit television imagery to a human operator.
for interpretation. A video image processor (VIP) system, namely Autoscope (Figure 2.9), consists of one or more cameras, a microprocessor-based computer for digitizing and processing the imagery, and software for interpreting the images and converting them into traffic flow data. Video image processor systems detect vehicles by analyzing the imagery from a traffic scene to determine changes between successive frames.

![Figure 2.9: Autoscope -Video Image Processor](www.Autoscope.com)

Advantages

i) Easy installation.

ii) Low maintenance.

Disadvantages

i) Vulnerability to viewing obstructions; inclement weather; shadows; vehicle projection into adjacent lanes; occlusion; day to-night transition; vehicle/road contrast; water; salt grime; cobwebs on camera lens that can affect performance.

ii) Susceptible to camera motion caused by strong winds.

iii) The installation of a video image processor requires 30 to 50-feet mounting height for optimum presence detection and speed measurement.
iv) A video image processor arrangement is cost effective only if many detection zones are required within the field of view of the camera

2.2.3 Video Image Processing Approaches

Video Image processing approaches are based on the video images of the traffic. They can be broadly classified into:

a) Knowledge based methods

b) Stereo-vision based methods

c) Motion based methods

a) Knowledge based methods

Knowledge based methods employ a prior knowledge to hypothesize vehicle locations in an image. Information about symmetry, color, shadow, corners, horizontal/vertical edges, texture, and vehicle lights is used to detect vehicles. A synopsis of Knowledge-based characteristics is discussed below:

i) Symmetry

Vehicle images observed from rear or frontal view are in general symmetrical in horizontal and vertical directions. This observation was used as a cue for vehicle detection in the early 90s. This approach is sensitive to noise when computing symmetry from intensity of a vehicle with homogeneous area on it. In [1], when searching for local symmetry, two issues must be considered carefully. First, a rough indication of where a vehicle is probably present is required. Second, even when using both intensity and edge maps, symmetry as a cue is still prone to false detections, such as symmetrical background objects or partly occluded vehicles.
ii) Color

Although few existing systems use color information to its full extent for Hypothesis Generation (HG), it is a very useful cue for obstacle detection, lane/road following, etc. Several prototype systems investigated the use of color information as a cue to follow lanes/roads or segment vehicles from background [2]. Similar methods could be used for HG, because non-road regions within the road area are potentially vehicles or obstacles. The lack of deploying color information in HG is largely due to the difficulties of color-based object detection or recognition methods for outdoor settings.

iii) Shadow

Using shadow information as a sign pattern for vehicle detection was discussed [3]. By investigating image intensity, it was found that the area underneath a vehicle is distinctly darker than any other areas on an asphalt paved road. A first attempt to deploy this observation was discussed in [4], though there was no systematic way to choose appropriate threshold values. The intensity of the shadow depends on the illumination of the image, which in turn depends on weather conditions. It should be noted that the assumption about the distribution of road pixels might not always consistent. For example, rainy weather conditions or bad illumination conditions will make the color of road pixels dark, causing this method to fail.

iv) Corners

Exploiting the fact that vehicles in general have a rectangular shape, Bertozzi et.al [5] proposed a corner-based method to hypothesize vehicle locations. Four templates, each of them corresponding to one of the four corners, were used to detect all the corners in an
image, followed by a search method to find the matching corners. For example, a valid upper-left corner should have a matched lower-right corner.

v) Vertical/Horizontal Edges

Different views of a vehicle, especially rear views, contain many horizontal and vertical structures (rear-window, bumper etc). Using constellations of vertical and horizontal edges has shown to be a strong cue for hypothesizing vehicle presence. However, an important issue to be addressed, especially in the case of on-line vehicle detection is how the choice of various parameters affects system robustness. These parameters include the threshold values for the edge detectors, the threshold values for picking the most important vertical and horizontal edges and the threshold values for choosing the best maxima (i.e., peaks) in the profile images. Although a set of parameter values might work perfectly well under some conditions, they might fail in other environments.

vi) Texture

The presence of vehicles in an image cause local intensity changes. Due to general similarities among all vehicles, the intensity changes follow a certain pattern, referred to as texture [6]. This texture information can be used as a cue to narrow down the search area for vehicle detection. Entropy was first used as a measure for texture detection. Another texture-based segmentation method suggested [6] used co-occurrence matrices. The co-occurrence matrix contains estimates of the probabilities of co-occurrences of pixel pairs under predefined geometrical and intensity constraints. Using texture for HG can introduce much false detection. For example, when we drive a car outdoor, especially in some downtown streets, the background is very likely to contain textures.
vii) Vehicle Lights

Most of the cues discussed above are not helpful for night time vehicle detection. It would be difficult or impossible to detect shadows, horizontal/vertical edges or corners in images obtained at night. Vehicle lights represent a salient visual feature at night. Cucchiara [7] used morphological analysis for detecting vehicle light pairs in a narrow inspection area.

b) Stereo-Vision based Methods

Stereo Vision means the perceptual experience of seeing solidity objects in 3 dimensions. There are two types of methods using stereo information for vehicle detection. First one uses disparity map, while the second method uses an anti-perspective transformation (Inverse Perspective Mapping (IPM)).

i) Disparity Map

The difference in the left and right images between corresponding pixels is called disparity. The disparities of all the image points form the so-called disparity-map. If the parameters of the stereo rig are known, the disparity map can be converted into a 3-D map of the viewed scene. However computing the disparity map, however, is very time consuming including complex computations

ii) Inverse Perspective Mapping

This method denotes an inversion under the additional constraint that inversely mapped points should lie on the horizontal plane. Assuming a flat road, Zhao et al. [8] used stereo vision to predict the image seen by the right camera using IPM given the left image: Specifically, they used the IPM to transform every point in the left image to world coordinates, and re-projected them back onto the right image, which were then compared
against the actual right image. In this way, they were able to find contours of objects above the ground plane. Instead of warping the right image onto the left image, Bertozzi et al. [9] computed the inverse perspective map of both the right and left images.

In general, stereo-vision based methods are accurate and robust only if the stereo parameters have been estimated accurately, which is really hard to guarantee in the on-road scenario. Since the stereo rig is on a moving vehicle, vibrations from car motion can shift the cameras while the height of the cameras can keep changing due to the suspension.

c) Motion based Methods

All the cues discussed so far use spatial features to distinguish between vehicles and background. Another important cue that can be used is the relative motion obtained using optical flow calculation. Optical flow information can provide strong information for HG. Approaching vehicles at an opposite direction produce a diverging flow, which can be quantitatively distinguished from the flow caused by the car ego-motion [10]. On the other hand, departing or overtaking vehicles produce a converging flow. Giachetti et al. [10] developed first-order and second-order differential methods and applied them to a typical image sequence taken from a moving vehicle along a flat and straight road. The results were discouraging [10]. Three factors causing poor performance were summarized as: (a) displacement between consecutive frames, (b) lack of textures and (c) shocks and vibrations. Given the difficulties faced by moving camera scenario, getting a reliable dense optical flow is not an easy task. Giachetti et al. [10] managed to re-map the corresponding points between two consecutive frames by minimizing a distance measure. Kruger et al. [11] estimated the optical flow from spatio-temporal derivatives of the grey
value image using a local approach. They further clustered the estimated optical flow to eliminate outliers. In contrast to dense optical flow, “sparse optical flow” utilizes image features such as corners [12], local minima and maxima [13], or “Color Blob” [14]. Although it can only produce a sparse flow, feature based method can provide sufficient information for HG. In contrast to pixel based optical flow estimation methods where pixels are processed independently, feature based methods utilize high level information. Consequently, they are less sensitive to noise.

In general, motion-based methods can detect objects based on relative motion information. Obviously, this is a major limitation, for example, this method cannot be used to detect static obstacles which can represent a big threat.

i) Temporal Thresholding

Vehicle detection based on the calculation of optical flow is computationally expensive and therefore not practicable for real-time system based on a low cost hardware.

Temporal Thresholding, where the moving pixels are identified by thresholding the temporal difference between the frames. Two approaches are followed for Frame Differencing, Inter-Frame Difference and Ref-Frame Difference.

1) Inter-Frame Difference

Here the object is detected by differencing two successive images. The temporal derivative of luminosity only is considered; a point is moving if its luminosity changes between two consecutive frames. These techniques have been proposed since many years with many further improvements such as accumulative difference [15] or with double-difference between three frames [11, 3]. The double difference is more precise in locating
real objects. Inter-frame difference is interesting, since is very fast in frame-rate application. The drawback is that often these techniques do not produce close object contours and therefore morphological operations (e.g. dilation) must be added or an edge closure guided by the edge gradient [16]. However, these techniques can be useful when objects' motion is mainly along a known direction, as for instance when a road is monitored with coming or outgoing vehicles.

2) Ref-Frame Difference

The method proposed [17] is based on the threshold of a difference image obtained by subtracting a reference frame to the current frame. The reference frame is refreshed choosing randomly 64 pixels that are substituted by the corresponding pixels in the current frame. In [18], the method is based on the difference between the histogram of a reference frame and the histogram of the current frame. If the difference is considerable, a vehicle is counted. A motion detection method based on the analysis of the difference frame between the captured frames and a reference frame is proposed [19]. This reference frame is build by an exponential temporal smoothing of the captured frames. The motion detection is made by comparing the intensity of the pixels in the difference frame with a threshold previously defined.

A motion detection technique that uses morphological edge detection (SMED - Separable Median Filter) and reference frame differencing is presented [20]. This technique obtains the edges of the current and the reference frame and subtracts them obtaining the edges of the moving objects.
3) Spatio-Temporal Slices Processing

The traditional approaches to video vehicle detection tend to formulate computational methodologies on the single frame, spatio-temporal slices provide rich information along a larger spatial and temporal scale to improve the precision of vehicle detection. This approach is based on the pattern analysis of Spatio-Temporal Slices.

The virtual-line-based algorithm and the area-based algorithm are two important and widely used solutions among various kinds of algorithms for Video Vehicle Detection (VVD). The virtual-line-based algorithm judges the vehicle presence with the luminance value changes of pixels in the virtual line as vehicles cross the virtual line which is set in the image. This algorithm has the advantages of high operation speed and strong real-time ability and disadvantages of low self-adaptive ability and inferior robustness [21]. The area-based algorithm judges vehicle presence with the pixel value changes in the area which is set in the image. This algorithm has low real-time efficiency in spite of the strong ability of self-adaptation and robustness [21]. To avoid disadvantages of both algorithms mentioned above, a novel VVD algorithm based on the pattern analysis of spatio-temporal slices (STS) was introduced. The newly developed algorithm uses rich information along a larger spatial and temporal scale to improve the precision of vehicle detection. Additionally, it efficiently restrains the influences of shadows, window reflection, tailgating vehicles, special vehicles, etc. on the accuracy of detection.

ii) Background Subtraction

Background subtraction is a widely used approach for detecting moving objects in videos from static cameras. The rationale in this approach is that detecting the moving objects from the difference between the current frame and a reference frame, often called
the “background image” or “background model.” The background image must be a representation of the scene with no moving objects and must be kept regularly updated so as to adapt to the varying luminance conditions and geometry settings.

The following simple approaches aim to maximize speed and limit the memory requirements to more sophisticated approaches aiming to achieve the highest possible accuracy under any possible circumstances. In background subtraction approach, detection occurs by comparing the incoming frame with a background model of the scene that is built by modeling the pixel intensity.

The equation for subtraction the current image with the background is given by:

\[ |I(p) - B(p)| > T(p) \Rightarrow M(p) = 1 \]  \hspace{1cm} (2.1)

\[ |I(p) - B(p)| < T(p) \Rightarrow M(p) = 0 \]  \hspace{1cm} (2.2)

where I is the current image, B is the background image. M is the object image and T is the threshold at time k and pixel p.

Several methods for performing background subtraction have been proposed namely:

1) Running Gaussian Average
2) Temporal Median Filter
3) Mixture of Gaussians
4) Kernel Density Estimation
5) Sequential Kernel Density
6) Co-occurrence of image variations
7) Eigen Backgrounds
1) Running Gaussian average

Wren et al. [22] proposed to model the background independently at each \((i,j)\) pixel location. The model is based on ideally fitting a Gaussian Probability Density Function (pdf) on the last \(n\) pixel values. In order to avoid fitting the pdf from scratch at each new frame time, 't' a running (or online cumulative) average is computed instead as follows:

\[
\mu_t = \alpha I_t + (1 - \alpha)\mu_{t-1}
\]

where \(I_t\) is the pixel's current value, \(\mu_t\) the previous average and \(\alpha\) is an empirical weight often chosen as a tradeoff between stability and quick update.

The model in [22] was proposed for intensity images, extensions can be made for multiple-component color spaces such as \((R,G,B), (Y,U,V)\) and others. Moreover, if real-time requirements constrain the computational load, the update rate of either \(\mu\) or \(\alpha\) can be set to less than that of the sample (frame) rate. However, the lower the update rate of the background model the less a system will be able to quickly respond to the actual background dynamic.

2) Temporal median filter

Cucchiara et al. [23] proposed to compute the median of special set of values containing the last \(n\), sub-sampled frames and \(w\) times the last computed median value. This combination increases the stability of the background model.

The main disadvantage of a median-based approach is that its computation requires a buffer with the recent pixel values. Moreover, the median filter does not accommodate for a rigorous statistical description and does not provide a deviation measure for adapting the subtraction threshold.
3) Mixture of Gaussians

Over time, different background objects are likely to appear at a same (i,j) pixel location. This is due to a permanent change in the scene's geometry, all the background subtraction models discussed so far will reflect the value of the current background object. However, sometimes the changes in the background object are not permanent and appear at a rate faster than that of the background update. A typical example is that of an outdoor scene with trees partially covering a building a same (i,j) pixel location will show values from tree leaves, tree branches, and the building itself.

In [24], Stauffer and Grimson describe the probability of observing a certain pixel value, \( x' \), at time \( t' \) by means of a mixture of Gaussian as:

\[
P(x_t) = \sum_{i=1}^{K} \omega_{i,t} \operatorname{N}(x_t - \mu_{i,t}, \sum_{ii})
\]

(2.4)

with each of the ‘K’ Gaussian distributions deemed to describe only one of the Observable background or foreground objects. In practical cases, ‘K’ is set to be between 3 and 5. Gaussians are multi-variable to describe red, green and blue values. If these values are assumed independent, the co-variance matrix, \( \sum_{ii} \) simplifies to diagonal. In addition, if the standard deviation for the three channels is assumed the same, it further reduces to a simpler, \( \sigma^2 I \),

At each \( t \) frame time, two problems must be simultaneously solved a) assigning the new observed value, \( x \) to the best matching distribution and b) estimating the updated model parameters. These concurrent problems can be solved by expectation maximization (EM) algorithm working on the buffer of the last \( n \) frames.
4) Kernel Density Estimation

An approximation of the background probability density function (pdf) can be given by the histogram of the most recent values classified as background values. However, as the number of samples is necessarily limited such an approximation suffers from significant drawbacks: the histogram might provide poor modeling of the true unknown pdf. In order to address such issues, Elgammal et al. in [25] have proposed to model the background distribution by a non-parametric model based on Kernel Density Estimation (KDE) on the buffer of the last ‘n’ background values. KDE guarantees a smoothed, continuous version of the histogram.

In [25], the background pdf is given as a sum of Gaussian kernels centered in the most recent ‘n’ background values $x_t$:

$$P(x_t) = \frac{1}{n} \sum_{t=1}^{n} \eta(x_t - x_t, \Sigma_t)$$

(2.5)

All the approaches model independently single pixel locations. However, it is intuitive that neighboring locations will exhibit spatial correlation in the modeling and classification of values. To exploit this property, various morphological operations have been used for refining the binary map of the classified foreground pixels. In [25] the same issue is addressed at the model level, by evaluating $P(x_t)$ also in the models from neighboring pixels and using the maximum value found in the comparison against ‘T’.

5) Sequential Kernel Density

Mean-shift vector techniques have recently been employed for various pattern recognition problems such as image segmentation and tracking [26, 27]. The mean shift vector is an effective gradient-ascent technique able to detect the main modes of the true pdf directly from the sample data with a minimum set of assumptions (unlike the
Stauffer-Grimson approach, the number of modes is unrestricted). However, it has a very high computational cost since it is an iterative technique and it requires a study of convergence over the whole data space. As such, it is not immediately applicable to modeling background pdfs at the pixel level.

Piccardi and Jan [28] propose some computational optimizations promising to mitigate the computational drawback. Moreover, in a recent paper from Han et al. [27], the mean-shift vector is used only for an off-line model initialization [28]. In this step, the initial set of Gaussian modes of the background pdf is detected from an initial sample set. The real-time model update is instead provided by simple heuristics coping with mode adaptation, creation, and merging. In their paper, Han et al. compared the pdf obtained with their method against that of a KDE approach over a 500-frame test video, finding a low mean integrated squared error in the order of $10^{-4}$. This justifies the name of sequential Kernel Density approximation (SKDA) that the authors gave to their method. Over the test video in [29], the number of modes showed to vary between 3 and 11, with an average value of 8.

6) Co-occurrence of image variations

Seki in [30] proposes a new approach beyond the idea of mere chronological averages by exploiting spatial co-occurrence of image variations. Their main statement is that neighboring blocks of pixels belonging to the background should experience similar variations over time. Although this assumption holds good for blocks belonging to a same background object (such as an area with tree leaves), it will evidently not hold for blocks at the border of distinct background objects (this is likely the cause of several false detections shown [31], appearing at the borders of different background objects).
In [30], it is not specified whether the learning phase should be repeated over time to guarantee model update. As this model is based on variations, it is likely to show a natural robustness to limited changes in the overall illumination level. However, a certain update rate would be needed to cope with more extended illumination changes.

7) Eigen backgrounds

The approach proposed by Oliver [31] is also based on Eigen value decomposition, but this time applied to the whole image instead of blocks. Such an extended spatial domain can extensively explore spatial correlation and avoid the tiling effect of block partitioning. The method in [31] can be summarized as follows:

a) Learning phase

i) A samples of ‘n’ images is acquired, each image with ‘p’ pixels; the average image, $\mu$, is then computed and all images mean-subtracted.

ii) The covariance matrix is computed and the best $M$ eigenvectors stored in an eigenvector matrix, $\phi_{mb}$ of size $M \times p$.

b) Classification phase

Every time a new image, $I$, is available, it is projected onto the eigen space as

$$I' = \phi_{mb}(I - \mu)$$

(2.6)

$I'$ is then back projected onto the image space as

$$I'' = \phi_{mb}^T I' + \mu$$

(2.7)

since the eigen space is a good model for the static parts of the scene, but not for the small moving objects, ‘$I''$ will not contain any such objects. Foreground points are eventually detected at locations where,

$$|I' - I''| > T$$

(2.8)
The above procedure can be subject to variations improving its efficiency.

8) Optical Flow

Optical flow is a technique to detect moving vehicles by segmenting dense optic flow fields into background and occlusion layers. The optical flow based algorithms extract a dense velocity field from an image sequence assuming that image intensity is conserved during the displacement. This conservation law is expressed by a spatiotemporal differential equation which is solved under additional constraints of different form. A recent survey of the optical flow techniques and a comparative empirical evaluation study of their performance can be found [32]. Recently, Quénot et al.[33] presented an optical flow algorithm based on a dynamic programming technique and tested the algorithm with PIV sequences.

Dynamic programming was originally used for searching the optimal matching between two one-dimensional patterns. In [33], it is extended to two dimensions: the global matching is searched that minimizes the L1 or L2 distance between two images. This is achieved by the Orthogonal Dynamic Programming (ODP) algorithm that slices the two images into properly selected sets of parallel overlapping strips. The corresponding strips are then matched as one-dimensional signals. This is repeated twice for two orthogonal slicing directions. This process is iterated in a multi-resolution way so as to refine the previously obtained matching. The number of iterations is a parameter of the algorithm and depends on the complexity of the velocity field. A few iterations are sufficient for relatively simple flows, such as the standard PIV sequences [34].

The algorithm in [33], yields a dense velocity field between any two images of a sequence, provided that the displacements between the two frames are not too large. The
method computes a velocity value in each pixel of the image, even though it may be less accurate near the image borders.

Both correlation methods and optical flow techniques imply matching performed for the whole image, which can be very time consuming for large images. When individual particles can be detected, and when there is no need to measure a velocity vector in each pixel of the image, feature based techniques are also applicable and may provide a faster solution.

2.3 Vehicle Tracking

After motion detection, Video Image Processing systems track moving objects from one frame to another frame in the video sequence. The existing tracking algorithms may be classified into four major categories:

a) Region-based tracking.

b) Active contour-based tracking.

c) Feature-based tracking and

d) Model-based tracking.

Active contour-based algorithms have received the most attention due to their compactness and mathematical validity.

The snake or active contour is a typical dynamic algorithm to extract object contours, which was presented by Kass, Witkin and Terzopoulos []. The GVF snake largely solves both the problems of the traditional snake’s initialization and its poor convergence to boundary concavities. An active contour model based on the gradient vector flow of the gray level and on a motion similarity measure was proposed by Yang
However, many snake algorithms cannot deal with object topological changes, and they require complex initialization. Fortunately, level-set theory [10] has turned out to be able to handle topological change with front propagation. Its initialization is so simple that the contour can even be initialized with the object itself.

The level-set method is a powerful tool for tracking the evolution of fronts propagating with curvature-dependent speed. [10] has some similarities to the classic snake algorithms: it consists of a moving contour, and it can be made responsive to user-specified parameters. It also has advantages over the classic snake: it is marker less, does not depend on a step size, handles sharp corners and handles changes in topology splits and merges. But it has one small limitation that usually its speed function depends only on the image gradient field. Location errors will appear sometimes when the tracking contour goes through a region that has the similar color to the object and they will affect the post-processing.

2.3.1 Methodologies in Object Tracking

Recognition of general three-dimensional objects from 2D images and videos is a challenging task. The common formation of the problem is essentially: given some knowledge of how certain objects may appear, plus an image of a scene possibly containing those objects, find which objects are present in the scene and where recognition is accomplished by matching features of an image and model of an object. The two most important issues that a method must address are the definition of a feature, and how the matching is found.
Object recognition methods can be classified according to a number of characteristics.

a) Appearance-Based Recognition.

b) Histograms

   i) Color histograms

   ii) Histogram comparison measures.

c) Probabilistic Interpretation (Probability density estimation).

a) Appearance Based Method

   The central idea behind appearance-based methods is the following. Having seen all possible appearances of an object, can recognition be achieved by just efficiently remembering all of them?

   Appearance Based Methods typically include two phases. In the first phase, a model is constructed from a set of reference images. The set includes the appearance of the object under different orientations, different illuminations and potentially multiple instances of a class of objects as shown in Figure 2.10.

   ![Figure 2.10: Two orientations of a object](image)

   In the second phase, “recall”, parts of the input image (sub images of the same size as the training images) are extracted by means of segmentation (by texture, color, motion). The recognition system then compares an extracted part of the input image with the reference images.
A major limitation of the appearance-based approaches is that they require isolation of the complete object of interest from the background. They are thus sensitive to occlusion and require good segmentation.

b) Histograms

i) Color Histograms

Swain and Ballard [76] proposed to represent an object by a color histogram. Objects are identified by matching histograms of image regions to histograms of a model image. While this technique is robust to object orientation, scaling and occlusion.

Given the tri stimulus R,G,B for each pixel, compute the 3D histogram for that image using the formula:

\[ H(R,G,B) = \# \text{(pixels with color}(R,G,B) \]  \tag{2.9} \]

Figure 2.11 and Figure 2.12 show a 3-D perspective plot of \( H(R,G,B) \) as a function of R, G and B.

(a) (b)

Figure 2.11: 3-D Plot

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
One component of the 3D color space is intensity. If a color vector is multiplied by a scalar, the intensity changes, but not the color itself. This means colors can be normalized by the intensity (Figure 2.13) is given by

\[ I = R + G + B \]  \hspace{1cm} (2.10)

Chromatic representation

\[ r = R / (R + G + B) \] \hspace{1cm} (2.11)

\[ g = G / (R + G + B) \] \hspace{1cm} (2.12)

\[ b = G / (R + G + B) \] \hspace{1cm} (2.13)

Figure 2.13: Chromatic representation
since \( r+g+b=1 \), only two parameters are necessary. Example one can use \( r \) and \( g \) and \( b \) can be obtained as follows:

\[
b = I - r - g
\]  

(2.14)

Merits
i) Invariant to object translations.

ii) Invariant to image rotations.

iii) Slowly changing for out of plane rotations.

iv) No perfect segmentation necessary. Histograms change gradually when part of the object is occluded.

v) Possible to recognize deformable objects.

Demerits
i) The pixel colors change with the illumination (color constancy problem).

ii) Not all objects can be identified by their color.

b) Histogram Comparison Methods

The requirements for obtaining histogram of objects by means of previous methods are generally impossible to achieve, because for example it is impossible to recognize objects in images taken in complete darkness. The challenge is then to develop a method with minimal constraints. An effective way for recognizing the objects in images taken in complete darkness is image histogram method (Figure 2.14). The decisive step in this technique is correctly matching corresponding image histograms in adjacent images. Intensity is one feature, which for the most part is independent of the motion of the objects and is thus especially well suited for matching. In these cases the image histogram can be used as matching only if the image-to-image differences are relatively
small. Represent each object by an image histogram. For recognizing the objects just compare the image histogram of test image with known objects as shown in Figure 2.14.

Figure 2.14: Object Histograms

The various histogram comparison methods for recognizing the objects are

i) Intersection.

ii) Euclidean Distance.

iii) Chi-Square.

i) Intersection

This method measures the common part of both histograms. The basic formula for recognizing the object using intersection comparison measure (shown in Figure 2.15) is given as:

\[ \cap (Q,V) = \sum \min(q_i, v_i) \quad (2.15) \]

\( q_i \) = histogram of test image and \( v_i \) =histogram of known object

ii) Euclidean Distance

Euclidean Distance one such method used for recognizing the object by using the image histograms of test image and knows objects respectively. This method mainly
focuses on the differences between the histograms. In this method all the cells are weighted equally. The Basic formula for calculating the Euclidean distance is given as:

\[ d(Q, V) = \sum (q_i - v_i)^2 \]  \hspace{1cm} (2.16)

iii) Chi-square

Another method for recognizing the object by using the image histograms of test image and known object is Chi-Square. The Basic formula for calculating the Euclidean distance is given as:

\[ \chi^2(Q, V) = \sum \frac{(q_i - v_i)^2}{q_i + v_i} \]  \hspace{1cm} (2.17)

This method tests whether the two distributions are different (or) not and the cells are not weighted equally and they are more robust to outliers.

iii) Probabilistic Recognition

Probability of object, \( O_n \), given measurement \( m_k \) is defined as:

\[ p(o_n|m_k) = \frac{p(m_k|o_n)p(o_n)}{p(m_k)} = \frac{p(m_k|o_n)p(o_n)}{\sum p(m_k|o_i)p(o_i)} \]  \hspace{1cm} (2.18)

where,

\( p(o_n) \) the a priori probability of object \( o_n \);

\( p(m_k) \) the a priori probability of measurement \( m_k \);

\( p(m_k/o_n) \) the probability density function of object \( o_n \);

Probabilistic Recognition Algorithm:

i) Build up histograms \( p(m_k/o_n) \) for each training object.

ii) Sample the test image to obtain \( m_k, k \in K \)

iii) Compute the probabilities for each training object as follows:
iv) Select the object with the highest probability or reject the test image if no object accumulates sufficient probability.

2.4. Summary

In this chapter, the classification of vehicle detection and tracking methods has been covered. On the whole, vehicle detection by means of intrusive sensors has the disadvantages namely: single detection zone, inconvenient and disruption of traffic for installation, degradation of pavements and roads due to maintenance, inaccurate detection over period of time. While non-intrusive sensors are: easy to install, provide multiple detection zones, low equipment maintenance, provide more traffic information. Non-intrusive type video based vehicle detection is more sophisticated than all other methods due to its ability to detect vehicles in all weather conditions by competitive approaches with good accuracy. The next chapter is concerned with the Vehicle Detection System and its equipment.
CHAPTER 3

SYSTEM SETUP AND EQUIPMENT

In this chapter, the system setup and equipment are discussed. First building of the system for implementation of two applications namely Automatic Work Zone Information (AWI) and Queue Detection (QD) Systems have been focused.

3.1 Components of the QD System and AWI System

The proposed systems consist of the following components:

a) Camera Unit
b) Digital Video Recorder
c) Battery Power Supply Unit
d) RF Transceiver Unit
e) Autoscope Interface Processor Unit
f) VMS Interface Processor

The block diagram of the system with the components is shown in Figure 3.1. The actual snap shot of the components are shown in Figure 3.2.
Figure 3.1: Block Diagram of the System

Figure 3.2: Snapshot of the components used in the system
3.1.1. Camera Unit

The camera unit selected for our system was based on the following criteria:

i) Resolution.

ii) Field of view.

iii) Illumination.

iv) Power supply and consumption.

The selected camera unit, PC219ZWPH High Resolution Weather Resistant 5-50mm Zoom Camera has the following specifications:

The wide range zoom vari-focal 5-50MM lens and 350 line, (480 lines HR model) 0.6 lux color camera combine to deliver a new level of performance, reliability, convenience and flexibility. DC driven auto-iris lens and built-in backlight compensation ensure fantastic video performance in almost all lighting conditions. Camera measures 6.1"H x 3.5"W x 7.87"D. (including integral sunshield/ rainhood). Runs on 12 volts DC and draws 130 milliamps. Specification of the PC219ZWPH High Resolution Weather Resistant 5-50mm Zoom Camera is provided in Table A.1 in appendix A.

3.1.2. Digital Video Recorder

A digital video recorder (DVR) or personal video recorder (PVR) is a device that records video in a digital format to a disk drive or other medium. The DVR unit selected for our system was based on the following criteria:

a) Resolutions and frames per sec. recording capabilities

b) Storage capacity for recordings.

c) Power supply and consumption.
The selected DVR unit, Triplex Stand-Alone 4 CH DVR - CDR4060 has the following specifications (provided in Table A.2 in appendix A):

a) High Resolution: 720 x 480
b) Real Time Display: 120 frames/sec
c) Recording Frame Rate: 30 frames/sec
d) Hard Disk Drive: 80GB
e) 4 Video Inputs
f) Event Recording by Alarm Sensor or Video Motion Detection
g) Programmable Recording Speed
h) 1 CH Audio In & Out
i) Built-in P/T/Z control, compatible with Pelco-D protocol

3.1.3 Battery Power Supply Unit

The battery power supply unit for the DVR, Camera, Autoscope and the Processors consists of two 12V deep-cycle batteries connected in parallel or serial to provide sufficient operation voltage and current. An analysis of current requirement for the system was conducted and two batteries in parallel were determined to be sufficient to provide the power for the system. As per calculations the two batteries would provide an un-interrupted power supply of 6 hrs before re-charging. Figure 3.3, shows the battery connections adopted in our implementation:
The most important consideration in buying a deep cycle battery is the Ampere-Hour (AH) or Reserve Capacity (or Reserve Minutes) rating that will meet our requirements. Most deep cycle batteries are rated in discharge rates of 100 hours, 20 hours, or 8 hours. The higher the discharge, the lower the capacity due to the Peukert Effect and the internal resistance of the battery. Reserve Capacity (RC) is the number of minutes a fully charged battery at 80°F (26.7°C) is discharged at 25 amps before the voltage falls below 10.5 volts. To convert Reserve Capacity (RC) to Ampere-Hours at the 25 amp rate, multiple RC by .4167. More ampere-hours (or RC) are better in every case. Specifications of Exide Deep Cycle Battery is provided in appendix A.

3.1.4 RF Transceiver Unit

The RF transceiver unit for the system consists of the following components: 1) RF Transmitter with yagi-pole antenna, 2) RF Receiver with yagi-pole antenna or base-station antenna. The RF module selected for the system was based on the range of communication and power consumption. The RF module implemented in our system was XStream-PKG-R 900 MHz, manufactured by Maxstream, INC.

The Digi XStream-PKG-R 900 MHz & 2.4 GHz stand-alone RF Modems provide outstanding range (up to 20 miles) in a low cost wireless solution. The modem is coupled with a DIP-switchable RS-232 / RS-422 / RS-485 interface board.
No configuration is necessary for out-of-box RF communications. The modem’s default configuration supports a wide range of data system applications. Advanced configurations can be implemented using simple AT or binary commands. Figure 3.4, shows product specification of Digi XStream-PKG-R 900 MHz stand-alone RF Modem.

a) Antenna:

High-gain antennas offer the maximum range available to our MaxStream RF module. The Yagi-antenna selected for our system is a 4 element, Gain (dBi) 8.1, Frequency Range (MHz) 902-928, etc. The specification and the picture of the antenna is provided in Table A.3 in appendix A. (Figure 3.4).

Figure 3.4: 4 Element Yagi Antenna

3.1.5 Communication Between Stations

System setup has two Autoscope sub-systems namely Remote-A and Remote-B and one VMS sub-system called Base Station. The communication between these stations is based on the following algorithms (Figure 3.5 and Figure 3.6).
START

Transmit "A"

Wait for 8 bit data from "Remote A" for 3 sec

Store the data

Wait for 2 sec

Transmit "B"

Wait for 8 bit data from "Remote B" for 3 sec

Store the data

Wait for 2 sec

Compare data received from Remote A and Remote B

UpdateVariable Message Sign Board

Figure 3.5: VMS Interface Processor Algorithm
3.1.6 Autoscope Interface Processor Unit

The Autoscope interface processor unit consists of a Basic Stamp© processor that interfaces with the DB15 Autoscope interface. The BASIC Stamp is a microcontroller developed by Parallax, Inc. which is easily programmed using a form of the BASIC programming language. The BASIC Stamp runs on 5 to 15 volts DC. The BS2-IC consumes 8 mA in running mode and 100 µA in sleep mode, not including any circuitry on the I/O pins. The BS2-IC measures 1.2" (30 mm) L x 0.62" (16 mm) W x 0.35" (9 mm) D. The BASIC Stamp II Carrier Board measures 2.8" (71 mm) L x 3.1" (79 mm) W x 0.6" (15 mm) D. The BASIC Stamp II uses a PIC16C57.

Microchip Technology Inc. The BASIC Stamp modules will work in 0 to 70°C temperatures with up to 70%, non-condensing humidity. After you write the code for your application, you simply connect the BASIC Stamp to the computer’s serial port or
USB, provide power to the BASIC Stamp and press ALT-R (DOS version) or CTRL-R (Windows version) within the appropriate BASIC Stamp editor to download your program into the BASIC Stamp’s EEPROM. As soon as the program has been downloaded successfully, it begins executing its new program from the first line of code. Figure 3.8, shows the diagram of the Basic Stamp and the carrier board that houses the processor.

Figure 3.7: a) Basic Stamp Module and b) Carrier Board with Serial interface

3.1.7 Autoscope Rackvision

Autoscope® RackVision is a machine vision vehicle detection technology. The RackVision vehicle detection system delivers the same high performance machine vision processing that is available in the Autoscope Solo Pro. The RackVision can be used with AIS black and white, color or CCTV cameras.

Autoscope RackVision™ has been designed to plug directly into standard US loop detector racks. It can be easily integrated into existing Autoscope Solo® communications installations. RackVision units can be networked together simply and automatically using built-in network and communications software and off-the-shelf Ethernet cables.
a) Features

i) Single camera video detection processor card and TS1 I/O in a 2-slot width footprint.

ii) Provides basic vehicle detection, traffic data measurement, and incident detection.

iii) Plugs into existing standard US loop detector racks.

iv) Stores traffic data in non-volatile memory.

v) Auto-selects NTSC or PAL video formats.

vi) Low power consumption.

vii) Self test on power-up.

viii) Local language support.

ix) IP addressable into Autoscope Solo® network, using sophisticated networking at high data rates.

x) LED indicates status for communications, valid video, data processing, and power.

xi) Detector I/O via front panel DB 15 connector.

xii) Optional video compression card.

b) Benefits

i) Cost-effective solutions for traffic management.

ii) Field proven accuracy.

iii) Easy to install and configure.

iv) Flexibility to meet a variety of detection and surveillance needs.

v) Expanded application opportunities.

vi) Connect to existing black and white or color Autoscope AIS or CCTV cameras.

vii) Up to 6 Rack Vision units and a power supply can easily fit into a standard US loop amplifier rack.

viii) Optional enclosure for shelf mounting.
c) Setup & Operation

Simple mouse and keyboard operations add, delete, or move up to 99 virtual detection zones. Customize detection to meet your requirements.

Detection zones for count, presence, speed, and incident detection applications. Real-time polling or stored data include: volume, occupancy, speed, five vehicle classes, and other traffic data over selected time periods. Status of incident and other alarms are available. Local output support for speed and classification unavailable, unless connected to a Detector Port Master Module. Easily assign detector outputs to interface with existing traffic control equipment.

Provides 8 outputs and 4 inputs via edge connector or front panel DB15 (Jumper selectable). This feature is used by our system to generate outputs to inform the VMS of traffic congestions and incidents.

3.1.8 Autoscope

a) Introduction

Vehicle detection by video cameras is one of the most promising new technologies for wireless large-scale data collection and implementation of advanced traffic control and management schemes such as vehicle guidance/navigation.

Autoscope, a new vehicle detection system can detect traffic in many locations (i.e., multiple spots) within the camera’s field-of-view. These locations are specified by the user in a matter of minutes using interactive graphics and can be changed as often as desired. This flexible detection placement is achieved by placing detection lines along or across the roadway lanes on a TV monitor displaying the traffic scene. Therefore, these detection lines are not physically placed in the pavement but only on the TV monitor. Every time a car crosses these lines, a detection signal (presence or passage) is generated by the device.
Because of this design, Autoscope can be installed without disrupting traffic operations. Furthermore, it is not locked to a particular detection configuration. The detection configuration can be changed manually or dynamically (i.e., by software as a function of traffic conditions). Because of these features, this video detection system leads to the advanced traffic surveillance and control applications. Autoscope calibration is shown in Figure 3.8.

b) Development of detector configuration

i) Calibration

Before calibrating an Autoscope, either determination of real-world distance between each of the calibration lines positioned on an image (Standard calibration) is required or a set of reference points should be loaded into configuration

ii) Standard Calibration

To determine the real-world reference point measurements, one of the following ways can be followed:

1. Go to the roadway and measure the distances between physical objects, such as lane markers, light posts, and other stationary, visible items. The physical objects are reference points.

2. Go to the roadway and create your own artificial ground markers between which to measure distances.

3. Work from blueprints of the roadway.

4. Use known distances as reference points. For example, the width of traffic lanes, the distances between lane hash marks, or the dimensions of concrete pads are already known.
Once the actual distances between the reference points on the ground are determined, the same positions and distances can be assigned to the gridlines.

Figure 3.8 shows the calibration lines, which can be placed and sized as desired to match the reference points previously identified on the ground and seen in a still image. While 10 calibration lines are available, it is necessary to actually position and specify location distances for only 5 of these 10. (At least 3 must come from one set of lines, Crosslane or Downlane, and 2 must come from the other set of lines). All the 10 can also be used.

The arrowhead in the upper, left-hand corner of the work area refers to the direction of travel of traffic in the Autoscope’s field of view. This arrowhead automatically orients itself to the direction of the average of the Crosslane lines. Can select which set of 5 lines corresponds to Crosslane and Downlane by using the check boxes in the Calibration dialog.

![Figure 3.8: Calibration of autoscope](image)

54

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
iii) Presence detector

Presence detectors identify the presence of a vehicle in the field of view. Their high accuracy in determining the presence of vehicles makes them ideal for signal-controlled intersection applications. Presence detectors which process an image, can be optimized for:

1. Vehicles moving in any direction: The detector will turn ON while a vehicle passes and remains under the detector.
2. Vehicles moving in only one direction: This can be used to eliminate detection of vehicles not traveling in the desired direction, or to detect vehicles traveling in the wrong direction.

Before making a Presence detector directional, field of view must be calibrated. Recommended size of the presence detector is 0.6 to 1.2 meters (2 to 4 feet) thick and 1.5 to 2 car lengths long. Typically, single Presence detector accuracy lies in the 95% to 98% range. By combining multiple detectors, Presence detector accuracy in excess of 99% can be achieved.

In Figure 3.9, the two directional Presence detectors are joined with the AND Detector Function and Presence Detector Configuration is shown in Figure 3.10.
iv) Presence detector parameters

1) Detector id number: Identifies the detector number and is not editable.

2) Detector Title: Uniquely identifies this detector for future reference.
3) Detection Parameters. Specifies detection conditions.

4) Background Refresh Rate (n sec): Specifies the guaranteed minimum time (for safety) an object may remain stationary before the MVP begins to consider it as background: For freeway applications, 30 to 60 seconds. For intersection applications, 20 to 600 seconds.

5) Night Reflection. Specifies whether the MVP compensates for roadway reflections that cause false detections. If this feature is not selected, light projected from a vehicle's headlights may turn ON a detector if the beam is sufficiently dense. The detector may then measure the vehicle length as both the beam plus the vehicle itself.

6) Orientation: Defines whether the Presence detector is a Crosslane or Downlane detector.
   i. Downlane: If the Presence detector is parallel to the lane, specifies a Downlane detector.
   ii. Crosslane: If the Presence detector is perpendicular to the lane, specifies a Crosslane detector.

7) Direction: Defines in which direction to detect traffic. Cannot specify direction until the field of view is calibrated.

8) Shadow Processing: Enables additional detection processing to prevent false vehicle detections caused by the shadows cast by vehicles in adjacent lanes.

9) No Shadow Processing: Turns off shadow processing. Select if shadows do not pose detection problems.
10) Indoor Lighting: Prevents false detections caused by shadows cast by vehicles in adjacent lanes when indoor lighting is used and there is no movement of the sun through the sky to change the shadow direction (such as in a tunnel).

11) Outdoor Lighting: Prevents false vehicle detections caused by shadows cast by vehicles in adjacent lanes. When you edit this parameter, the software shows a Sun icon. The movement of the sun through the sky is taken into account

v) Speed detector

Speed detectors (shown in Figure 3.13) generate these types of traffic information:

1) Vehicle speeds: either in kilometers-per-hour or miles-per-hour. Speed is always reported as an integer.

2) Vehicle lengths: Length is reported in decimeters but the variable is passed in meters to the Data Collector.

3) Vehicles that are classified into five categories based upon the measured length. The categories are Classes A through E.

4) The Speed detector uses a simple graphic and the Speed detector length is automatically sized based on the maximum expected speed set up by the user.

In order for a Speed detector to function properly, it must be placed on the upstream side of the associated Count detector. The Speed detector processes traffic data only after the associated Count detector changes state from ON to OFF.

Speed detectors appear as a trapezoid on the detector layout are shown below. The trapezoid shape must be placed upstream from the associated Count detector. Once a vehicle passes through the Speed detector zone and the associated Count detector turns
ON, the Speed detector automatically calculates the vehicle speed. Speed Detector Configuration is shown in Figure 3.12.

Figure 3.11: Speed Detector

Figure 3.12: Speed Detector Configuration
vi) Speed detector parameters

1) Detector id number: Identifies the detector number. Not editable. Example: 106.

2) Detector Title: Uniquely identifies this detector for future reference.

3) Max. Report Speed (n km/h, mi/h): Specifies the highest speed the detector reports.

4) Min. Classification Speed (n km/h, mi/h): Specifies the lowest speed a vehicle may travel and still be measured and classified.

5) Min. Report Speed (n km/h, mi/h): Specifies the lowest speed the detector reports.

6) Speed Calibration Adjustment: Specifies a speed measurement adjustment factor that compensates for traffic traveling at different road surface heights, such as over and under bridges. In the field of view, higher elevation traffic appears to move faster than lower elevation traffic traveling at the same speed. The measured speed is multiplied by the speed measurement adjustment factor to get a new adjusted reported speed.

7) Min. Vehicle Length (n meters/feet): If the value in the option is not zero, then the minimum call placed by the detector is the minimum length times the measured speed. This option prevents the occurrence of very short detector actuations.

8) Trigger Enable Speed (n km/h, mi/h): The Speed detector allows you to use one detector or a combination of detectors to act as a lane discriminator (to avoid crosslane occlusion) when connected to a Speed detector. Typically, downlane Presence detectors are used as lane discriminators.

9) Speed Trap Spacing (n meters/feet): Specifies a spacing distance that is used only when detector outputs (for example, using the I/O) are required to simulate the
output of two inductive loops. The measured vehicle speed and the spacing distance are used to calculate the delay between two output pulses.

10) Apply Median Speed Filter: When selected, reports only the median speed of the current measured speed and the last 4 valid speeds measured (5 total). This filter is particularly useful in freeway applications where the speeds of closely following vehicles are very similar. It also removes outlier values.

11) Vehicle Classification Thresholds (meters/feet): Defines the threshold values for vehicle classification.

12) Vehicle Length Adjustment: Specifies multiplication factor to adjust how lengths are viewed.

13) A through E: Defines minimum, maximum, and intermediate vehicle lengths for five categories.

14) Occupancy Normalization: Enables the MVP to correct Speed detector occupancy (ON time) for comparison with other types of visual detection sensors, such as inductive loops.

15) Detector Length to Emulate (n meters/feet): Specifies length of emulated detection zone.

16) Detector Length Normalization Factor: Corrects any biases found in the occupancy outputs (such as constantly 10% high).

17) Show Speed: Shows measured speed on video display.

18) Show Class: Shows vehicle classification on video display.

19) Visible: When checked, this causes the overlay graphics for the Speed detectors to be visible on the video that is output by the MVP. When unchecked, the overlay
graphics for the Speed detectors are not visible on the video that is output by the MVP.

20) Use as default for new detectors: Each time when a new detector of this type is created, the selections made in this dialog are applied to the new detector as defaults.

vii) Count detector

Count detectors (shown in Figure 3.13) perform the vehicle detection processing (i.e., the detectors show whether or not there is a vehicle under the detector). Count detectors compile traffic volume statistics. Volume is the sum of the vehicles detected during a time interval specified when:

1) Set preferences.

2) Defined the file configuration.

There are no restrictions on how to orient count detectors in the field of view. Count detectors appear as straight lines that are most often drawn perpendicular to a lane of traffic. Count detector configuration is shown in Figure 3.14.

Figure 3.13: Count Detector
viii) Count detector parameters

1) Detector id number: Identifies the detector number and is not editable.

2) Detector Title: Uniquely identifies this detector for future reference.

3) Detection Parameters: Specifies detection conditions.

4) Background Refresh Rate (n sec): Specifies the guaranteed minimum time (for safety) an object may remain stationary before the MVP begins to consider it as background.

5) Night Reflection: Specifies whether the MVP compensates for roadway reflections that cause false detections.
7) Approaching: Defines traffic as moving towards the MVP.

8) Receding: Defines traffic as moving away from the MVP.

9) Shadow Direction: Prevents false vehicle detections caused by shadows cast by vehicles in adjacent lanes. When you edit this parameter, the software shows a Sun icon.

10) Indoor Lighting: Prevents false detections caused by shadows cast by vehicles in adjacent lanes when indoor lighting is used and there is no movement of the sun through the sky to change the shadow direction (such as in a tunnel).


12) Outdoor Lighting: Prevents false vehicle detections caused by shadows cast by vehicles in adjacent lanes. When you edit this parameter, the software shows a Sun icon. The movement of the sun through the sky is taken into account.

13) Morning: Defines the list of possible directions from which the morning shadows approach the detector.

14) Afternoon: Defines the list of possible directions from which the evening shadows approach the detector.

15) Night: Defines the list of possible directions from which the night shadows approach the detector.

16) Turn Off Shadow Processing at Midday: Selects whether to stop processing shadows at midday, when shadows are normally at the lowest level. Midday is defined as the period when the sun is higher than n degrees. The number of degrees (n) is based on the Latitude/Longitude settings for the detector file.
defined as the period when the sun is higher than n degrees. The number of
degrees (n) is based on the Latitude/Longitude settings for the detector file.

17) Visible: When checked, this causes the overlay graphics for the Count detectors to
be visible on the video that is output by the MVP. When unchecked, the overlay
graphics for the Count detectors are not visible on the video that is output by the
MVP.

18) Use as default for new detectors: Each time you create a new detector of this type,
the selections made in this dialog are applied to the new detector as defaults.

ix) Boolean detector function

A Boolean Detector Function (Detector Function shown in Figure 3.15) combines the
normal outputs of two or more detectors into one customized output. Typically, a single
detector performs its function, such as counting, when a vehicle enters its zone. The
detector registers a simple ON or OFF output to the MVP. However, when two or more
detectors are combined into a Detector Function, Boolean logic relays the conditions that
must be present before the grouping generates an ON output. Boolean detector function
configuration is shown in Figure 3.16.
Figure 3.15: Boolean detector function

<table>
<thead>
<tr>
<th>Detector Title</th>
<th>Detector Function 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation Type</td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NAND</td>
</tr>
<tr>
<td></td>
<td>NOR</td>
</tr>
<tr>
<td></td>
<td>AND</td>
</tr>
<tr>
<td></td>
<td>OR</td>
</tr>
<tr>
<td></td>
<td>AND</td>
</tr>
<tr>
<td></td>
<td>NAND</td>
</tr>
<tr>
<td></td>
<td>NOR</td>
</tr>
<tr>
<td></td>
<td>AND</td>
</tr>
<tr>
<td></td>
<td>OR</td>
</tr>
<tr>
<td></td>
<td>NAND</td>
</tr>
<tr>
<td></td>
<td>NOR</td>
</tr>
<tr>
<td></td>
<td>AND</td>
</tr>
<tr>
<td></td>
<td>OR</td>
</tr>
<tr>
<td></td>
<td>NAND</td>
</tr>
<tr>
<td></td>
<td>NOR</td>
</tr>
<tr>
<td></td>
<td>AND</td>
</tr>
<tr>
<td></td>
<td>OR</td>
</tr>
<tr>
<td></td>
<td>NAND</td>
</tr>
<tr>
<td></td>
<td>NOR</td>
</tr>
<tr>
<td></td>
<td>AND</td>
</tr>
<tr>
<td></td>
<td>OR</td>
</tr>
</tbody>
</table>

Clearing Distance: 0.0 seconds
Contrast Force: 0.0 seconds
Contrast Phase: 0.0 seconds
Controller Phase: 0.0 seconds
Permissive Controller Phase: 0.0 seconds

- Initial State DFF
- Force cell during poor scene contrast
- Enable contrast check at night (tunnels)

Figure 3.16: Boolean detector function configuration

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
x) Detector function parameters

1) Detector id number: Identifies the detector number and is not editable.

2) Detector Title: Uniquely identifies this detector for future reference.

3) Operation Type: Combines separate detectors into one customized Detector Function. Select one of these Operation Types for the Detector Function to operate properly. When the conditions set by the Boolean Operation selected, are present, the Detector Function turns ON:

4) OR: When you select OR, the Detector Function turns ON when at least one of the member detectors of the Detector Function is ON.

5) AND: When you select AND, the Detector Function turns ON only when all of the member detectors of the Detector Function are ON.

6) NAND: When you select NAND, the Detector Function turns ON when at least one of the member detectors of the Detector Function are OFF.

7) NOR: When select NOR, the Detector Function turns ON only when all of the member detectors of the Detector Function are OFF.

8) M of N: When you select M of N, the Detector Function turns ON only when a defined number M, of the N member detectors, is ON.

9) Output: Output types are predefined to follow a certain course of action based on the controller phase input.

10) Type: Describes the operating criteria for each detector Output Type. When selected a particular output type, the MVP monitors traffic for conditions that match the output type. If conditions match, the MVP sends the Detector Function call to the traffic controller (when assigned to an I/O output).
11) Extend Time (n seconds): Defines how long Detector Function stays ON after the Detector Function logic indicates an OFF condition. For example, when the traffic conditions no longer exist, the Detector Function continues to generate the call for an additional number of seconds.

12) Delay Time (n seconds): Defines how long the Detector Function should wait before it turns ON after the ON conditions have been met. For example, if the Detector Function turns ON, we may want the Detector Function to wait another 10 seconds before turning ON. If a Detector Function turns OFF before the delay time expires, the Detector Function does not turn ON.

13) Clearing Distance (n feet/meters): Defines the length (in meters or feet) of a roadway dilemma zone. This zone is typically the distance between a Speed detector and the other side of an intersection, as seen on the detector layout. A Dilemma Zone is the area where traffic hazards are most likely to occur. The Detector Function automatically calculates the length of time it takes for a vehicle to clear this zone.

14) Controller Phase/Load Switch: Instructs the Detector Function which input phase to monitor for this Detector Function.

15) Permissive Controller Phase/Load Switch: A permissive controller phase is one where the vehicle driver may at his discretion take an action if it is safe to do so. The driver is not depending on the controller to ensure that his action is safe.

16) Locking: Sets an output call to last the duration of an input phase status.

17) Initial State OFF: When this flag is set, the startup state of the Detector Function and the state of any TS1 and/or TS2 outputs assigned to the detector are modified...
from the normal behavior. During startup, the Detector Function state will be OFF until any one of the following conditions are met by any of its member detectors.

18) Contrast: Defines an override for normal Detector Function when the scene contrast falls below a certain level.

19) Visible: When checked, this causes the overlay graphics for the Detector functions to be visible on the video that is output by the MVP. When unchecked, the overlay graphics for the Detector Functions are not visible on the video.

20) Use as default for new detector functions: Each time a new detector of this type is created, the selections made in this dialog are applied to new detector as defaults.

3.1.9 System Integration

The diagram of the sub-system that detects and transmits the information to the RF base-station is shown in Figure 3.17. The sub-system consists of the following components: 1) Camera unit, 2) Power supply unit, 3) Autoscope Rackvision, 4) Autoscope interface processor unit and 5) RF transmitter module. Flow chart of operation is presented in Figure 3.18.

![Autoscope sub-system block diagram](image)

Figure 3.17: Autoscope sub-system block diagram

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
3.1.9 VMS Interface Processor

The VMS Interface Processor is also a BASIC Stamp is a microcontroller developed by Parallax, Inc. which is easily programmed using a form of the BASIC programming language. The VMS interface processor receives the signal transmitted from the Autoscope and displays the appropriate message on the Variable Message Sign (VMS). The diagram of the sub-system is shown in Figure 3.19. Flow chart of operation is presented in Figure 3.20.

Figure 3.18: Autoscope sub-system operation flow
Figure 3.19: VMS sub-system block diagram

RF Module
Receives Data from the Autoscope sub-system

VMS Interface Processor
Converts Autoscope signals to VMS Data

Message Display

Figure 3.20: VMS sub-system operation flow
3.1.10. Improved VMS Interface Processing

The communication between VMS and Basic Stamp is not successful because of inefficiency of Basic Stamp transmitting data in required hexadecimal format. This drawback has been overcome by replacing Basic Stamp with Visual Basic Application. The new Improved VMS subsystem can be viewed as below (Figure 3.21). Snapshot of Developed VB application is shown in Figure 3.22.

![Figure 3.21: Improved VMS sub-system block diagram](image)

![Figure 3.22: Snapshot of VB Application](image)
3.2 Harris Corner based Vehicle Detection System

The Setup comprises of Embedded XP Board, Hauppauge Frame Grabber and Camera Unit (Figure 3.23). The Visual C++ Application is converted into .exe and downloaded onto Embedded XP Board.

![Diagram of Stand Alone Video Processor](image)

Figure 3.23: Stand Alone Video Processor

3.2.1 OLYMPUS Windows XP Embedded Development Kit

Arcom’s Windows XP Embedded Development Kit is designed to provide a complete environment allowing software engineers to concentrate on the development of their application. Kit includes a Windows XP Embedded operating system license label and allows developing, downloading and debugging Windows XP Embedded applications. The Windows XP Embedded operating system has been configured to run on the Arcom OLYMPUS board. The Windows XP Embedded License label is attached to the Compact Flash card. Board Connection are shown in Figure 3.24.

Arcom's OLYMPUS Windows XP Embedded Development Kit comprises of the following items:

i) OLYMPUS-M256-F0 processor board (fitted with Intel 1GHz Socket370 Pentium III processor, heat sink and fan).

ii) OLYMPUS Windows XP Embedded Development Kit Support CD.
iii) OLYMPUS Windows XP Embedded Development Kit Installation CD.

iv) Windows XP Embedded download utilities floppy disk (with licensed copy of Data light ROM-DOS and Sockets).

v) +5V(8A) and +12V(1.7A) power supply (100V-240VAC input).

vi) US, UK or European power cord.

vii) CRT VGA adapter cable.

viii) Keyboard/mouse cable.

ix) ATA/66 IDE cable (80 conductors).

x) 2 x serial adapter cables.

xi) Parallel port adapter cable.

xii) Floppy disk drive cable.

xiii) 2 x USB adapter cable.

xiv) Ethernet adapter cable.

xv) Reset button and cable.

xvi) Power button and cable.

xvii) OLYMPUS board power cable.

xviii) Power extension cable.

xix) PS/2 mouse.

xx) DVI Interface board (DVI-1).

xxi) 1M DVI-D Cable.

xxii) 1M Ethernet crossover cable.

xxiii) OLYMPUS Windows XP Embedded Development Kit Quickstart Manual.

xxiv) Development Kit storage case.
xxv) 512M Bytes CompactFlash.

Additional items are

i) TFT flat panel display and Touchscreen

ii) NEC 6.5’ Color TFT flat panel display (NL6448BC20-08)

iii) Backlight inverter module

iv) Flat panel cable assembly

v) Arcom TSC1 (Touchscreen controller)

Figure 3.24: Embedded Board Connections
(Courtesy: www.arcom.com)

3.2.2. Hauppauge USB-Live FrameGrabber

Hauppauge USB-Live (Figure 3.25) is an external device which connects your VCR, camcorder or video camera to PC or Laptop.
i) It digitizes video using high quality 4:2:2 video sampling. And we can work on applications while simultaneously watching video in a resizable window.

ii) Connects to any S-Video or composite video source and plugs into PC or laptop's USB port.

iii) USB-Live's video digitizer turns analog video into digital video, and then sends 30 digital images per second over the USB (1.1 or 2.0) bus.

iv) It is a video digitizer and can be used with Internet videoconferencing applications such as Microsoft's NetMeeting, CU-SeeMe and more! Videoconference.

v) With exclusive WinTV SnapShot application, we can quickly save video images! Capture 24-bit still video still images in TIFF, JPEG, BMP in color or black/white. Also great for web design.


Minimum system Requirements to Install WinTV USB

i) PC with Pentium processor (500MHz MMX minimum recommended).

ii) Microsoft® Windows® 98SE/2000/Me/XP.

iii) A Sound card with LINE IN.

iv) USB 1.1 or 2.0 port.
3.3 Summary

The system setup and involved equipment has been studied in detail in this chapter. Next chapter is concerned with an algorithm developed in Visual Basic to handle vehicle detection and Tracking.
CHAPTER 4

HARRIS CORNER DETECTION

In this chapter, Harris Corner Detection algorithm with interest points consideration is proposed for vehicle detection and tracking. Harris Corner Method (HCM) is a popular detector due to its strong invariance to rotation, scale, illumination variation and image noise. The Harris algorithm involves following steps: (1) Computation of image derivatives in X and Y directions; (2) Formation of autocorrelation matrix $C(x, \sigma, \tilde{\sigma})$, also called second moment gradient matrix; (3) Computation of eigenvalues of second moment matrix and comparison with a certain threshold, indicating a corner. Vehicle detection is basis for vehicle tracking and mid-point of detected corner points of a vehicle is taken as reference and its displacement in successive frames is calculated. The traffic parameter, speed of a vehicle is determined based on the displacement of midpoint of corner points.

Advantages of Harris Corner detection are: robust to noise, clutter, invariance to image transformations and robust to illumination changes. Limitations are incapable of full frame rate operation and intensive computation.

4.1 Harris Corner Algorithm

The Harris corner detector is based on the local auto-correlation function of a signal, where the local auto-correlation function measures the local changes of the signal with
patches shifted by a small amount in different directions. A discrete predecessor of the Harris detector was presented by Moravec [2]; where the discreteness refers to the shifting of the patches. A vehicle is detected by a set of interest points obtained by this algorithm. The detected vehicles are tracked using mid-point of their corners.

4.1.1 Derivation of Harris Algorithm

Given a shift \((\Delta x, \Delta y)\) and a point \((x, y)\), the auto-correlation function is defined as,

\[
c(x, y) = \sum_w [I(x_i, y_i) - I(x_i + \Delta x, y_i + \Delta y)]^2
\]  

(4.1)

where \(I(x_i, y_i)\) denotes the image function and \((x_i, y_i)\) are the points in the window \(W(\text{Gaussian})\) centered on \((x, y)\).

The shifted image is approximated by a Taylor expansion truncated to the first order terms,

\[
I(x_i + \Delta x, y_i + \Delta y) \approx I(x_i, y_i) + [I_x(x_i, y_i)I_y(x_i, y_i)] \frac{\Delta x}{\Delta y}
\]  

(4.2)

where \(I_x(x_i, y_i)\) and \(I_y(x_i, y_i)\) denote the partial derivatives in \(x\) and \(y\) respectively.

Substituting approximations in the equation (4.1) yields,

\[
c(x, y) = \sum_w [I(x_i, y_i) - I(x_i + \Delta x, y_i + \Delta y)]^2
\]  

(4.3)

\[
= \sum_w \left( I(x_i, y_i) - I(x_i, y_i) + [I_x(x_i, y_i)I_y(x_i, y_i)] \frac{\Delta x}{\Delta y} \right)^2
\]  

(4.4)

\[
= \sum_w \left( -[I_x(x_i, y_i)I_y(x_i, y_i)] \frac{\Delta x}{\Delta y} \right)^2
\]  

(4.5)

\[
= \sum_w \left( [I_x(x_i, y_i)I_y(x_i, y_i)] \frac{\Delta x}{\Delta y} \right)^2
\]  

(4.6)

\[
= \left[ \frac{\Delta x \Delta y}{\sum_w I_x(x_i, y_i)^2 \sum_w I_y(x_i, y_i)^2} \right] \left[ \sum_w I_x(x_i, y_i)I_y(x_i, y_i) \right]^2
\]  

(4.7)
\[
= [\Delta x \; \Delta y] C(x, y) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}
\]

where matrix \( C(x, y) = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \) captures the intensity structure of the local neighborhood.

Let \( \lambda_1, \lambda_2 \) be the eigenvalues of matrix \( C(x, y) \). The eigenvalues form a rotationally invariant description. There are three cases to be considered:

i) If both \( \lambda_1 \) and \( \lambda_2 \) are small: the local auto-correlation function is flat (i.e., little change in \( c(x, y) \) in any direction), the windowed image region is of approximately constant intensity.

ii) If one eigenvalues is high and the other low: local auto-correlation function is ridge shaped, then only local shifts in one direction (along the ridge) cause little change in \( c(x, y) \) and significant change in the orthogonal direction, this indicates an edge.

iii) If both eigenvalues are high: local auto-correlation function is sharply peaked, then shifts in any direction will result in a significant increase, this indicates a corner.

A corner is thus indicated by both \( \lambda_1 \) and \( \lambda_2 \) being large and a flat image region by \( \lambda_1 \) and \( \lambda_2 \) being small. In order to avoid explicit eigen decomposition of matrix \( C(x, y) \), it is attractive to use trace, \( Tr(C) \) and determinant, \( Det(M) \), as given below:

\[
Tr(M) = \alpha + \beta = a + b \quad (4.9)
\]
\[
Det(M) = \alpha \beta = ab - c^2 \quad (4.10)
\]

The following formulation for corner response is considered:

\[
R = Det(C) - kTr^2(C) \quad (4.11)
\]

80
where $R$ is known as Harris corner measure.

4.1.2 Determination of Harris Corners in an Image

i) Compute the image derivatives in $X$ and $Y$ directions viz., $l_x = \frac{\partial I(x,y)}{\partial x}$ and $l_y = \frac{\partial I(x,y)}{\partial y}$ respectively, where $I(x,y)$ is the image intensity at $(x, y)$. This is computed by using differential of Gaussian kernel of standard deviation $\sigma$.

ii) Form the autocorrelation matrix. This matrix $C(x, \sigma, \bar{\sigma})$ averages derivatives in a window around a point $x$. A Gaussian $G(\bar{\sigma})$ is used for weighing:

$$C(x, \sigma, \bar{\sigma}) = G(\bar{\sigma}) \otimes \begin{pmatrix} l_x^2(X, \sigma) & I_x I_y(X, \sigma) \\ I_x I_y(X, \sigma) & l_y^2(X, \sigma) \end{pmatrix}$$ (4.12)

In the above equation, $\bar{\sigma}$ is called the integration scale and $\sigma$ is called the differentiation scale. This matrix is also called as the second moment gradient matrix. $X$ is a Harris corner if, $Det(c) - \alpha \cdot trace^2(C) > \text{threshold}$ where the threshold is generally chosen between 1400 and 1600. $\alpha$ is a parameter which is chosen between 0.04 to 0.06.

iii) The eigenvalues of the second moment matrix at any point indicate the variance in $X$ and $Y$ directions at that point. Thus, the last step ensures that both the eigenvalues are greater than a certain threshold, indicating a corner.

4.1.3 Steps Involved in Detection and Tracking

i) Frame Capture.

ii) Preprocessing of frame.

iii) Smoothing.

iv) Color Conversion.

v) Implementation of Harris Corner algorithm.
vi) Separation of Vehicles.

i) Frame Capture.

Frames are captured by using Hauppauge frame grabber which can deliver up to 29.5 frames/sec. Frame rate is the measurement of the frequency (rate) at which an imaging device produces unique consecutive images called frames. The term applies equally well to computer graphics, video cameras, film cameras, and motion capture systems. Frame rate is most often expressed in frames per second (fps), or simply hertz (Hz).

Frame rates are used in synchronizing audio and pictures, whether film, television or video. In motion pictures and television, the frame rates are standardized by the Society of Motion Picture and Television Editors (SMPTE). SMPTE Time Code frame rates of 24, 25 and 30 frames per second are common, each having uses in different portions of the industry. The professional frame rate for motion pictures is 24 frames per second and for television, 30 frames per second (in the U.S.).

In computer video streams, the frame rate describes playback rates for AVI and QuickTime movies. The video playback rate for an AVI or QuickTime movie directly relates to the perceived smoothness of its playback. The higher the number of frames playing per second, the smoother the video playback appears to the user. Lower rates result in a choppy playback. (As a reference point, film uses 24 frames per second to allow the viewer to perceive smooth playback.) Several factors affect the actual frame rate got on your computer. For example, PC processor or graphics hardware may only be capable of playing 10-15 frames per second without acceleration.

In developing motion pictures, television and video, frame rate information is used as a reference for audio signals. The recorded signal includes information about location in
time using a 24-hour clock, and individual frame numbers. This signal is used to synchronize multiple audio and video machines during the recording and editing process. Using a master synchronizing device, the operator can issue location commands from a central machine and all slave machines follow the master.

ii) Preprocessing of Frame

It is sometimes of interest to process a single sub-region of an image leaving other regions unchanged. This is commonly referred to as region-of-interest (ROI) processing.

The captured frame from the frame grabber is of size 320*240 pixels. The frame consists of unwanted objects outside the region of interest (ROI). To enhance the processing performance, the frame is cropped to ROI specified by a rectangle of size 270*150. Theoretically, the size of image area has been reduced by 47.2% which improves the processing time by nearly the same amount.

Original size of frame = 320*240
New cropped frame = 270*150

iii) Smoothing

Smoothing is a process by which data points are averaged with their neighbors in a series, such as a time series or image. This (usually) has the effect of blurring the sharp edges in the smoothed data. Smoothing is sometimes referred to as filtering, because smoothing has the effect of suppressing high frequency signal and enhancing low frequency signal.

Example:

Here is a set of data, made out of random numbers that will be used as a pretended time series, or a single line of data from one plane of an image. The numbers were
generated with matlab, by creating 40 successive random numbers from a normal distribution (Figure 4.1).

![Figure 4.1: Sample data generated with Matlab](image_url)

The Gaussian kernel:

The 'kernel' for smoothing, defines the shape of the function that is used to take the average of the neighboring points. A Gaussian kernel is a kernel with the shape of a Gaussian (normal distribution) curve. Here is a standard Gaussian, with a mean value of 0 and a sigma (population standard deviation) of 1 shown in Figure 4.2.

![Figure 4.2: Gaussian kernel for smoothing](image_url)
In the standard statistical way, the width of the Gaussian shape is defined in terms of sigma. However, when the Gaussian is used for smoothing, it is usual to describe the width of the Gaussian with another related measure, the Full Width at Half Maximum (FWHM).

The FWHM is the width of the kernel, at half of the maximum of the height of the Gaussian. Thus, for the standard Gaussian above, the maximum height is approx. 0.4. The width of the kernel at 0.2 (on the Y axis) is the FWHM. As $x = -1.175$ and 1.175 when $y = 0.2$, the FWHM is 2.35.

Smoothing with a Kernel:

For each data point, a new value that is some function of the original value at that point and the surrounding data points is generated. With Gaussian smoothing, the function that is used is Gaussian curve.

Using a Gaussian with FWHM of 4 units on the x axis, the Gaussian kernel average for 14th data point is generated. First move the Gaussian shape to have its centre on 14 on the x axis (shown in Figure 4.3). In order to make sure that an overall scaling of the values after n smoothing is not done, then divide the values in the Gaussian curve by the total area under the curve, so that the values add up to one.
Take the values of resulting function (from the y axis), at each of the points in the data (the x axis). Thus generating Gaussian function values for 13 12 11 etc, and 15 16 17 etc. This gives a discrete Gaussian as shown in Figure 4.4.

In fact the Gaussian values for 12,13,14,15 and 16 are: 0.1174, 0.1975, 0.2349, 0.1975, 0.1174 and the data values for the same points are: 1.0645, 0.3893, 0.3490, -0.6566, -0.1946. Then multiply the Gaussian values by the values of data, and sum the results to get the new smoothed value for point 14. Thus, the new value for point 14 is ...
+... Then store this new smoothed value for future use, and move on, to $x = 15$, and repeat the process, with the Gaussian kernel now centered over 15. If this is done for each point, eventually the smoothed version of original data will be obtained (Figure 4.5).

![Figure 4.5: Smoothed value of original data](image)

Other Kernels:

Any shape for the kernel can be used, e.g. square wave (Figure 4.6). This would have the effect of replacing each data point with a straight average of itself and the neighboring points.

![Figure 4.6: Square wave Gaussian kernel](image)
Smoothing in 2-Dimension:

Smoothing in two dimensions follows simply from smoothing in one dimension. This time the Gaussian kernel is not a curve, but a cone. Figure 4.7 shows Gaussian kernel when placed over the central point of a plane. Figure 4.8 shows Gaussian kernel when discrete values for each pixel in the image are used. Then multiplying the values of the kernel by the data in the image, to get the smoothed value for that point, and doing the same for every point on the image.

Figure 4.7: Gaussian Kernel

Figure 4.8: Gaussian Kernel for discrete values of each pixel
The procedure is the same for 3D data, except the kernel is rather more difficult to visualize, being something like a sphere with edges that fade out, as the cone fades out at the edges in the 2D case.

In fact, it turns out that generation of this 2D and 3D versions of the kernel for the computations are not required, because the same result by applying the full 2 or 3D kernel can also be obtained if simply a one dimensional smooth sequentially in the 2 or 3 dimensions is applied. Thus, for 2 dimensions, first smooth in the x direction, and then smooth the x-smoothed data, in the y direction.

iv) Color Conversion

Color is the brain's reaction to a specific visual stimulus. Color can be precisely measured by its spectral power distribution (the intensity of the visible electro-magnetic radiation at many discrete wavelengths) this leads to a large degree of redundancy. The reason for this redundancy is that the eye's retina samples color using only three broad bands, roughly corresponding to red, green and blue light. The signals from these color sensitive cells (cones), together with those from the rods (sensitive to intensity only), are combined in the brain to give several different "sensations" of the color. These sensations have been defined by the CIE and are quoted from Hunt's book "Measuring Color".

1) Brightness: the human sensation by which an area exhibits more or less light.

2) Hue: the human sensation according to which an area appears to be similar to one, or to proportions of two, of the perceived colors red, yellow, green and blue.

3) Colorfulness: the human sensation according to which an area appears to exhibit more or less of its hue.
4) Lightness: the sensation of an area’s brightness relative to a reference white in the scene.

5) Chroma: the colorfulness of an area relative to the brightness of a reference white.

6) Saturation: the colorfulness of an area relative to its brightness.

The tri-chromatic theory describes the way three separate lights, red, green and blue, can match any visible color – based on the eye’s use of three color sensitive sensors. This is the basis on which photography and printing operate, using three different colored dyes to reproduce color in a scene. It is also the way that most computer color spaces operate, using three parameters to define a color.

a) Color space

A color space is a method by which one can specify, create and visualize color. One can define a color by its attributes of brightness, hue and colorfulness. A computer may describe a color using the amounts of red, green and blue phosphor emission required to match a color. A printing press may produce a specific color in terms of the reflectance and absorbance of cyan, magenta, yellow and black inks on the printing paper.

A color is thus usually specified using three co-ordinates, or parameters. These parameters describe the position of the color within the color space being used. They do not tell what the color is, that depends on what color space is being used.

Different color spaces are better for different applications, for example some equipment has limiting factors that dictate the size and type of color space that can be used. Some color spaces are perceptually linear, i.e. a 10 unit change in stimulus will produce the same change in perception wherever it is applied. Many color spaces, particularly in computer graphics, are not linear in this way.
Some color spaces are intuitive to use, i.e. it is easy for the user to navigate within them and creating desired colors is relatively easy. Other spaces are confusing for the user with parameters with abstract relationships to the perceived color. Finally, some color spaces are tied to a specific piece of equipment (i.e. are device dependent) while others are equally valid on whatever device they are used.

b) Types of color spaces

RGB (Red Green Blue):

This is an additive color system based on tri-chromatic theory. Often found in systems that use a CRT to display images. RGB is easy to implement but non-linear with visual perception. It is device dependent and specification of colors is semi-intuitive. RGB is very common, being used in virtually every computer system as well as television, video etc.

CMY (K) (Cyan Magenta Yellow (Black)):

This is a subtractive based color space and is mainly used in printing and hard copy output. The fourth, black, component is included to improve both the density range and the available color gamut (by removing the need for the CMY inks to produce a good neutral black it is possible to used inks that have better color reproductive capabilities). CMY (K) is fairly easy to implement but proper transfer from RGB to CMY (K) is very difficult (simple transforms are, to put it bluntly, simple). CMY (K) is device dependent, non-linear with visual perception and reasonably unintuitive.

HSL (Hue Saturation and Lightness):

This represents a wealth of similar color spaces, alternative names include HSI (intensity), HSV (value), HCI (chroma / colorfulness), HVC, TSD (hue saturation and...
darkness) etc. Most of these color spaces are linear transforms from RGB and are therefore device dependent and non-linear. Their advantage lies in the extremely intuitive manner of specifying color. It is very easy to select a desired hue and then modify it slightly by adjustment its saturation and intensity.

The supposed separation of the luminance component from chrominance (color) information is stated to have advantages in applications such as image processing. However the exact conversion of RGB to hue, saturation and lightness information depends entirely on the equipment characteristics. Failure to understand this may account for the sheer numbers of related but different transforms of RGB to HSL, each claimed to be better for specific applications than the others.

YIQ, YUV, YCbCr, YCC (Luminance - Chrominance):

These are the television transmission color spaces, sometimes known as transmission primaries. YIQ and YUV are analogue spaces for NTSC and PAL systems respectively while YCbCr is a digital standard.

This color spaces separate RGB into luminance and chrominance information and are useful in compression applications (both digital and analogue). These spaces are device dependent but are intended for use under strictly defined conditions within closed systems.

c) Color conversions

RGB to CMY:

\[
Cyan = 1 - Red \quad (4.13)
\]

\[
Magenta = 1 - Green \quad (4.14)
\]

\[
Yellow = 1 - Blue \quad (4.15)
\]
CMY to RGB:

\[
\begin{align*}
\text{Red} &= 1 - \text{Cyan} \quad (4.16) \\
\text{Green} &= 1 - \text{Magenta} \quad (4.17) \\
\text{Blue} &= 1 - \text{Yellow} \quad (4.18)
\end{align*}
\]

CMY to CMYK:

\[
\begin{align*}
\text{Black} &= \text{minimum} (\text{Cyan}, \text{Magenta}, \text{Yellow}) \quad (4.19) \\
\text{Cyan} &= \frac{\text{Cyan} - \text{Black}}{1 - \text{Black}} \quad (4.20) \\
\text{Magenta} &= \frac{\text{Magenta} - \text{Black}}{1 - \text{Black}} \quad (4.21) \\
\text{Yellow} &= \frac{\text{Yellow} - \text{Black}}{1 - \text{Black}} \quad (4.22)
\end{align*}
\]

CMYK to CMY:

\[
\begin{align*}
\text{Cyan} &= \text{minimum}(1, \text{Cyan} \ast (1 - \text{Black}) + \text{Black}) \quad (4.23) \\
\text{Magenta} &= \text{minimum}(1, \text{Magenta} \ast (1 - \text{Black}) + \text{Black}) \quad (4.24) \\
\text{Yellow} &= \text{minimum}(1, \text{Yellow} \ast (1 - \text{Black}) + \text{Black}) \quad (4.25)
\end{align*}
\]

d) Grayscale Image

A grayscale (or gray level) image is simply one in which the only colors are shades of gray. The reason for differentiating such images from any other sort of color image is that less information needs to be provided for each pixel. In fact a `gray' color is one in which the red, green and blue components all have equal intensity in RGB space, and so it is only necessary to specify a single intensity value for each pixel, as opposed to the three intensities needed to specify each pixel in a full color image.

Often, the grayscale intensity is stored as an 8-bit integer giving 256 possible different shades of gray from black to white. If the levels are evenly spaced then the
difference between successive gray levels is significantly better than the gray level resolving power of the human eye.

Grayscale images are very common, in part because much of today's display and image capture hardware can only support 8-bit images. In addition, grayscale images are entirely sufficient for many tasks and so there is no need to use more complicated and harder-to-process color images.

v) Vehicle Identification

The corner points obtained from Harris algorithm represent vehicles in all lanes. The sample video chosen has 4 lanes of traffic flowing in a direction away from camera. The separation of vehicles from one lane to other is done using calibrated reference lines dividing the traffic lanes. The calibration lines are of two types, vertical and horizontal. Vertical calibration lines are from equations (4.26) to (4.35) and horizontal calibration lines are equations (4.36) and (4.37).

\[
\begin{align*}
9x + 4y - 360 &< 0 \quad (4.26) \\
9x + 4y - 360 &> 0 \quad (4.27) \\
5x + 4y - 600 &< 0 \quad (4.28) \\
5x + 4y - 650 &> 0 \quad (4.29) \\
15x + 16y - 2750 &< 0 \quad (4.30) \\
15x + 16y - 2925 &> 0 \quad (4.31) \\
5x + 7y - 1300 &< 0 \quad (4.32) \\
5x + 4y - 1350 &> 0 \quad (4.33) \\
x + 2y - 390 &< 0 \quad (4.34) \\
x + 2y - 390 &> 0 \quad (4.35)
\end{align*}
\]
\[2x - 25y + 750 < 0 \quad (4.36)\]
\[2x - 25y + 2500 > 0 \quad (4.37)\]

The equations (4.27) and (4.28) correspond to lane 1, (4.29) and (4.30) correspond to lane 2, (4.31) and (4.32) correspond to lane 3, (4.33) and (4.34) correspond to lane 4.

vi) Vehicle Tracking

At this point of time, vehicles are detected by means of Harris method. To find the traffic parameters, vehicles are tracked using their corner points' mid-point. The vehicle's corner points are averaged to find the mid-point. The same methodology is applied to next successive frame, and the displacement of midpoint is calculated to determine the speed of the vehicle.

4.1.4 Harris Corner based Vehicle detection algorithm

i) STEP 1: Capture a frame from camera using Hauppauge USB-Live frame grabber and get the time stamp.

ii) STEP 2: Crop the image by choosing Region of Interest (ROI). Here Stationary objects are removed by restricting the image only to roads and traffic.

iii) STEP 3: Smooth the image by using Gaussian filter. Gaussian smoothing is preferred because the noise or the nature of the object observed might be of a Gaussian probable form.

iv) STEP 4: Convert the Color image into Grayscale image. The grayscale image serves as a input to Harris corner detection method.

v) STEP 5: Obtain corner points on the objects using Harris Corner method. Corner points are used as a basis for determining traffic parameters and are produced by Harris algorithm.
vi) STEP 6: Separate the obtained corner points into different lanes. The output of Harris method comprises of corner points (on the vehicles) of all the vehicles in different lanes. To distinguish one vehicle corner points from the other, first corner points of vehicles in same lane are separated from other.

vii) STEP 7: Find the vehicles in each Lane. Corner points in the same lane are further classified, corresponding to different vehicles.

viii) STEP 8: Repeat all the above steps for next frame and find the displacement of corresponding corner points.

ix) STEP 9: With reference to calibrated points on the road in the screen space, and the elapsed time between captured frames, find the traffic parameters of the vehicles.

4.1.5 Flow Chart

Figure 4.9, shows the flow chart of proposed algorithm for detection and tracking. The flow starts with vehicle detection and ends up with calculation of traffic parameter, speed of a vehicle.
Figure 4.9: Flow Chart

START

GRAB A FRAME

CROP THE FRAME

SMOOTH THE FRAME

CONVERT COLOR TO GREYSCALE

FIND CORNER POINTS OF VEHICLES

FIND VEHICLES IN EACH LANE

IF NOT FIRST FRAME

Yes

FIND TRAFFIC PARAMETERS

CONTINUE

Yes

STOP

No

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
4.2 Summary

This chapter covered the concept of Harris Corner algorithm for vehicle detection and tracking using the corner points mid-point as a basis. The next chapter would discuss the evaluation criteria, performance analysis of developed algorithm, followed by results.
CHAPTER 5

RESULTS AND CONCLUSION

5.1 Introduction to Methodology

In previous chapter, the concept of Harris corner detection algorithm has been discussed. In this chapter the implementation of Harris Corner detection algorithm for measurement of traffic parameters like speed, count etc will be discussed. In the first phase of detection, the Harris corners of the vehicle are determined, which represent the outline of vehicle. The midpoint of detected harris corners are calculated and is tracked in the consecutive frames to measure the linear displacement. The vehicle detection is the fundamental step of tracking for calculating traffic parameters like speed, count etc.

5.2 Performance Issues

The performance of the proposed algorithm is determined by calculating traffic parameters like speed, count. The term "False Detection" plays an important in determining algorithm’s efficiency. False detection is defined as detection of a vehicle in the absence of vehicle. The possible factors for this detection may be due to vehicle headlights, occlusion etc. Harris corner algorithm is robust to illumination factor because of adopted corners detection concept rather than background subtraction phenomenon.

99
5.3 Data Set

The performance of proposed algorithm has been evaluated using traffic videos recorded on three locations namely I-15, I-515 and Tropicana Ave, Las Vegas. Currently the evaluation is performed on I-15 traffic video data set. The data set comprises of seven clips of 2 minutes each recorded on Feb 10, 2007 during 2240 hrs and 2254 hrs; other video data sets will also be used soon. The pattern of recording data in these locations is shown below (Table 5.1).

<table>
<thead>
<tr>
<th>Location</th>
<th>Day/Night</th>
</tr>
</thead>
<tbody>
<tr>
<td>I-15 (Las Vegas)</td>
<td>Night</td>
</tr>
<tr>
<td>I-515 (Las Vegas)</td>
<td>Day (Afternoon)</td>
</tr>
<tr>
<td>Tropicana Ave (Las vegas)</td>
<td>Day (Morning)</td>
</tr>
</tbody>
</table>

Table 5.1: Data set recording locations and lighting conditions.

5.4 Experimental Setup

Video being the source of information, acts as a input to system. Traffic parameters namely speed and count are the outputs of the system. The Video is fed through Hauppauge frame grabber to the computer. The developed software application is self-executable and can be installed on any computer (min 600Mhz processor and a USB port). The functioning of the proposed system can be classified into three phases.

i) Calibration phase.

ii) Vehicle Detection phase.

iii) Speed Detection/ Count Detection phase.
5.4.1 Calibration Phase

Calibration is a process by which the conversion of road-space geometry is mapped into screen-space geometry. Developed VC++ application can be used for processing any traffic video irrespective of location. The calibration lines are of two types: horizontal and vertical. This pair of lines represents Detection Zone. The user has a provision to choose Detection Zone by means of inputting calibration lines coordinates at the command prompt. The horizontal calibration lines can be seen in Figure 5.2. The vertical calibration lines are shown in Figure 5.3. For better performance, large number of calibration lines can be used and are shown as modified vertical calibration lines in Figure 5.4. Vertical and Horizontal calibration lines together are shown in Figure 5.5.
5.4.2 Vehicle Detection Phase

The captured frame is of size 320*240. With the aim of achieving better processing performance, the frame has been cropped to 270*160, removing all the unwanted
stationary objects in the video. The noise in the frame is removed by using Gaussian filter (as discussed in Chapter 4). As a vehicle is represented by a set of corner points, first Harris corners on the vehicles in a frame are obtained and are shown with black spots in Figure 5.6. Figure 5.7 shows the detected Harris corner points represented by black spots with blue colored bounding rectangles.

![Figure 5.6: Detected Harris Corner Points on Vehicles Before Lane Separation](image)

![Figure 5.7: Harris Points with Rectangles on Vehicles Before Lane Separation](image)

Now the points obtained correspond to all vehicles in the frame. To distinguish corner points of one vehicle to other, first corner points of vehicles are grouped into separate lanes (indicating the vehicles in a particular lane), then individual vehicles using
detection zone, represented by red spots in lane1 in Figure 5.8 and green spots in lane2 in Figure 5.9.

Upon obtaining the Harris corners on a vehicle, the midpoint of corners is calculated by averaging the corners of that particular vehicle. Suppose \( c_1(x_1,y_1), c_2(x_2,y_2), c_3(x_3,y_3), \ldots c_n(x_n,y_n) \) are the pixel locations of corners of a vehicle, then the midpoint \( M(a,b) \) is calculated by

\[
M\left(a = \frac{x_1 + x_2 + x_3 + x_4 \ldots x_n}{n}, \ b = \frac{y_1 + y_2 + y_3 + y_4 \ldots y_n}{n}\right)
\] (5.1)
Midpoint of corner points of vehicle are calculated and tracked in consecutive frames. Figure 5.10 and Figure 5.11 show the midpoint of corner points of a vehicle in lane 2 in two consecutive frames.

5.4.3 Speed Calculation

One of the important goals of the proposed algorithm is determination of speed of vehicle. The calculation of speed is based on midpoint of corners of the vehicle. When a vehicle enters the detection zone, the speed detection module is activated and its corners midpoint is tracked through successive frames until it leaves detection zone. Figure 5.12 shows frame \( i \), in which a vehicle enters detection zone and its corners midpoint is shown.
with a red spot. The same vehicle in the next frames (i + 1) and (i+2) are shown in Figure 5.13 and Figure 5.14.

Figure 5.12: Frame i showing a Vehicle entered Detection Zone

Figure 5.13: Frame (i + 1) showing a Vehicle in Detection Zone

Figure 5.14: Frame (i + 2) showing a Vehicle in Detection Zone
The displacement between corner midpoints of a vehicle in successive frames is calculated as follows: If $M_1(a_1,b_1)$ and $M_2(a_2,b_2)$ are midpoints of corners of a vehicle in consecutive frames, then the linear displacement ($d$) between them is calculated by

$$d = \sqrt{(b_2 - b_1)^2 + (a_2 - a_1)^2} \quad (5.3)$$

To measure the speed of a vehicle, the midpoint displacement in $\text{pixels/sec}$ (Screen Space) is transformed into road space ($\text{Miles/hr}$) using the following transformation (Figure 5.15): Suppose $x$ is the displacement in $\text{pixels/msec}$ between two consecutive frames, then

$$\text{Displacement in } \frac{\text{miles}}{\text{hr}} = \frac{x \times 1000 \times 60 \times 60}{5280} \quad (5.4)$$

Figure 5.15: Conversion from Screen space to Road space geometry

5.4.4 Count Detection

Vehicles are counted when they enter detection zone (defined by vertical and horizontal calibration lines). Calibration horizontal line1 and line3 correspond to start and end of detection zone. Figure 5.16 shows a vehicle in lane2 which is about to enter
detection zone, vehicle in lane3 left the detection zone, a vehicle in lane4, still in
detection zone. Figure 5.17 shows a vehicle in lane2 in detection zone.

Figure 5.16: Vehicle towards detection zone in lane 2.

Figure 5.17: Vehicle in detection zone in lane 2.

5.5 Evaluation

The proposed system was evaluated using I-15 traffic video recorded during February
10, 2007 between 2240 hrs and 2254 hrs with reference to Autoscope. In Autoscope, data
logging pertaining to vehicular traffic parameters can be achieved by means of data
polling. In the same manner, developed application logs the vehicular pertaining data to a
text file specified by a path in the program.
5.6 Results

Employing the same video set as a data source for Autoscope and Corner Harris Midpoint based VC++ application, the results are shown below:

Table 5.2: Comparison of the CHM method and Autoscope for vehicle speed and count for Datasets 1 through 7.

<table>
<thead>
<tr>
<th>DATA SET 1</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>LANE</td>
<td>SPEED</td>
<td>COUNT</td>
<td>ERROR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CHM</td>
<td>AUTOSCOPE</td>
<td>CHM</td>
<td>AUTOSCOPE</td>
<td>ACTUAL</td>
<td>CHM</td>
</tr>
<tr>
<td>Lane1</td>
<td>62</td>
<td>67.3</td>
<td>18</td>
<td>22</td>
<td>18</td>
<td>0</td>
</tr>
<tr>
<td>Lane2</td>
<td>65.5</td>
<td>63.19</td>
<td>22</td>
<td>27</td>
<td>21</td>
<td>1</td>
</tr>
<tr>
<td>Lane3</td>
<td>63.6</td>
<td>65.4</td>
<td>29</td>
<td>34</td>
<td>28</td>
<td>1</td>
</tr>
<tr>
<td>Lane4</td>
<td>65.8</td>
<td>60.19</td>
<td>41</td>
<td>36</td>
<td>34</td>
<td>7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DATA SET 2</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>LANE</td>
<td>SPEED</td>
<td>COUNT</td>
<td>ERROR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CHM</td>
<td>AUTOSCOPE</td>
<td>CHM</td>
<td>AUTOSCOPE</td>
<td>ACTUAL</td>
<td>CHM</td>
</tr>
<tr>
<td>Lane1</td>
<td>62.5</td>
<td>62.1</td>
<td>22</td>
<td>21</td>
<td>21</td>
<td>1</td>
</tr>
<tr>
<td>Lane2</td>
<td>63.1</td>
<td>61.4</td>
<td>29</td>
<td>28</td>
<td>29</td>
<td>0</td>
</tr>
<tr>
<td>Lane3</td>
<td>64.7</td>
<td>61.3</td>
<td>41</td>
<td>36</td>
<td>32</td>
<td>9</td>
</tr>
<tr>
<td>Lane4</td>
<td>63.5</td>
<td>61.25</td>
<td>37</td>
<td>28</td>
<td>26</td>
<td>11</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DATA SET 3</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>LANE</td>
<td>SPEED</td>
<td>COUNT</td>
<td>ERROR</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>CHM</td>
<td>AUTOSCOPE</td>
<td>CHM</td>
<td>AUTOSCOPE</td>
<td>ACTUAL</td>
<td>CHM</td>
</tr>
<tr>
<td>Lane1</td>
<td>61.9</td>
<td>72</td>
<td>18</td>
<td>21</td>
<td>18</td>
<td>0</td>
</tr>
<tr>
<td>Lane2</td>
<td>64.2</td>
<td>62.18</td>
<td>22</td>
<td>27</td>
<td>21</td>
<td>1</td>
</tr>
<tr>
<td>Lane3</td>
<td>65.3</td>
<td>64.9</td>
<td>35</td>
<td>20</td>
<td>23</td>
<td>12</td>
</tr>
<tr>
<td>Lane4</td>
<td>66.8</td>
<td>62.4</td>
<td>32</td>
<td>29</td>
<td>27</td>
<td>5</td>
</tr>
</tbody>
</table>

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
### DATA SET 4

<table>
<thead>
<tr>
<th>LANE</th>
<th>SPEED</th>
<th>COUNT</th>
<th>ERROR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CHM</td>
<td>AUTOSCOPE</td>
<td>CHM</td>
</tr>
<tr>
<td>Lane1</td>
<td>63.8</td>
<td>64</td>
<td>17</td>
</tr>
<tr>
<td>Lane2</td>
<td>64.6</td>
<td>60.5</td>
<td>31</td>
</tr>
<tr>
<td>Lane3</td>
<td>65.1</td>
<td>62.3</td>
<td>35</td>
</tr>
<tr>
<td>Lane4</td>
<td>66.7</td>
<td>60.1</td>
<td>35</td>
</tr>
</tbody>
</table>

### DATA SET 5

<table>
<thead>
<tr>
<th>LANE</th>
<th>SPEED</th>
<th>COUNT</th>
<th>ERROR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CHM</td>
<td>AUTOSCOPE</td>
<td>CHM</td>
</tr>
<tr>
<td>Lane1</td>
<td>63.8</td>
<td>63.8</td>
<td>15</td>
</tr>
<tr>
<td>Lane2</td>
<td>65.1</td>
<td>64.4</td>
<td>23</td>
</tr>
<tr>
<td>Lane3</td>
<td>64</td>
<td>67.2</td>
<td>31</td>
</tr>
<tr>
<td>Lane4</td>
<td>63.1</td>
<td>57.4</td>
<td>37</td>
</tr>
</tbody>
</table>

### DATA SET 6

<table>
<thead>
<tr>
<th>LANE</th>
<th>SPEED</th>
<th>COUNT</th>
<th>ERROR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CHM</td>
<td>AUTOSCOPE</td>
<td>CHM</td>
</tr>
<tr>
<td>Lane1</td>
<td>62</td>
<td>67.8</td>
<td>13</td>
</tr>
<tr>
<td>Lane2</td>
<td>63.3</td>
<td>65.7</td>
<td>20</td>
</tr>
<tr>
<td>Lane3</td>
<td>64.13</td>
<td>65.06</td>
<td>32</td>
</tr>
<tr>
<td>Lane4</td>
<td>66.6</td>
<td>59.8</td>
<td>30</td>
</tr>
</tbody>
</table>

### DATA SET 7

<table>
<thead>
<tr>
<th>LANE</th>
<th>SPEED</th>
<th>COUNT</th>
<th>ERROR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CHM</td>
<td>AUTOSCOPE</td>
<td>CHM</td>
</tr>
<tr>
<td>Lane1</td>
<td>62.5</td>
<td>61.3</td>
<td>14</td>
</tr>
<tr>
<td>Lane2</td>
<td>62.7</td>
<td>60</td>
<td>27</td>
</tr>
<tr>
<td>Lane3</td>
<td>64.17</td>
<td>61.9</td>
<td>38</td>
</tr>
<tr>
<td>Lane4</td>
<td>64.53</td>
<td>54.4</td>
<td>38</td>
</tr>
</tbody>
</table>

110

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Table 5.3: Cumulative comparison of the CHM method and Autoscope for vehicle speed and count.

<table>
<thead>
<tr>
<th>LANE</th>
<th>SPEED CHM</th>
<th>SPEED AUTOSCOPE</th>
<th>COUNT CHM</th>
<th>COUNT AUTOSCOPE</th>
<th>COUNT ACTUAL</th>
<th>ERROR CHM</th>
<th>ERROR AUTO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lane1</td>
<td>62.6</td>
<td>65.5</td>
<td>117</td>
<td>124</td>
<td>112</td>
<td>5</td>
<td>12</td>
</tr>
<tr>
<td>Lane2</td>
<td>64</td>
<td>62.3</td>
<td>174</td>
<td>186</td>
<td>171</td>
<td>3</td>
<td>15</td>
</tr>
<tr>
<td>Lane3</td>
<td>64.4</td>
<td>63.7</td>
<td>241</td>
<td>219</td>
<td>197</td>
<td>44</td>
<td>22</td>
</tr>
<tr>
<td>Lane4</td>
<td>65.3</td>
<td>59.3</td>
<td>250</td>
<td>221</td>
<td>208</td>
<td>42</td>
<td>13</td>
</tr>
</tbody>
</table>

Table 5.4: Comparison of the CHM method and Autoscope for vehicle speed.

<table>
<thead>
<tr>
<th>SPEED</th>
<th>CHM</th>
<th>AUTOSCOPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SET 1</td>
<td>64.075</td>
<td>62.7</td>
</tr>
<tr>
<td>SET 2</td>
<td>64.225</td>
<td>64.02</td>
</tr>
<tr>
<td>SET 3</td>
<td>63.45</td>
<td>61.51</td>
</tr>
<tr>
<td>SET 4</td>
<td>64.55</td>
<td>65.37</td>
</tr>
<tr>
<td>SET 5</td>
<td>65.05</td>
<td>61.725</td>
</tr>
<tr>
<td>SET 6</td>
<td>64</td>
<td>63.2</td>
</tr>
<tr>
<td>SET 7</td>
<td>64</td>
<td>64.59</td>
</tr>
<tr>
<td>SET 8</td>
<td>63.475</td>
<td>59.4</td>
</tr>
<tr>
<td></td>
<td>64.103</td>
<td>62.814</td>
</tr>
</tbody>
</table>
Figure 5.18: Graphical Comparison of the CHM method and Autoscope for vehicle speed.

5.7 Discussion

The advance warning ITS system was developed that includes many off-the-shelf video based vehicle detection system. The video based detection system used requires calibration and fine tuning of configuration parameters for accurate results. Therefore, an in-house video based vehicle detection system was developed using the Conner Harris method (CHM) to eliminate the need of complex calibration and contrasts modifications. The performances of CHM and Autoscope are compared for vehicle speed and count. Provided below is a summary of the performance comparisons:

i) The performance of CHM is better when compared to Autoscope with respect to speed. The CHM provides an average speed of 64 mph compared to 62 mph determined by Autoscope. Earlier speed test using radar devices indicate that the
Autoscope determined speeds 5mph less than the actual speed. Therefore, CHM provides a better accuracy for speed than Autoscope.

ii) The performance of CHM is better than Autoscope for vehicle counts in Lanes 1 and 2. But for Lanes 3 and 4 the count values degrades significantly. This is due to following fact:

As the video capturing camera is installed on the light pole in the median, the direction of traffic flow is not parallel to y-axis of screen-space coordinate system; hence partial traffic in one lane is also present in next right lane from Camera’s point of view due to shadowing of vehicles. Lane1 and Lane2 are not much affected, since they are closer to camera. Autoscope performed better for this type of video, since it adopts background subtraction method rather than Corner point concept for vehicle detection.

5.8 Future Work

The proposed algorithm calculates traffic parameters namely speed and count. Other traffic parameters like volume, flow, and speed alarms can be calculated. The concept of “Lane changes” can be employed to enhance the performance of the system. Variations in the brightness of input video can be considered to design system more robust.
APPENDIX A

Table A.1: Specification of the PC219ZWPH High Resolution Weather Resistant 5-50mm Zoom Camera

<table>
<thead>
<tr>
<th>Specification</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizontal Resolution</td>
<td>480 Lines</td>
</tr>
<tr>
<td>Illumination</td>
<td>0.6 ~ 0.8 Lux/F1.8</td>
</tr>
<tr>
<td>Image Sensor</td>
<td>1/3&quot; CCD Sensor Interline</td>
</tr>
<tr>
<td>Power Requirements</td>
<td>12VDC</td>
</tr>
<tr>
<td>Power Consumption</td>
<td>320mA at 12V DC</td>
</tr>
<tr>
<td>Video Format</td>
<td>NTSC</td>
</tr>
<tr>
<td>Pixels</td>
<td>492 (V) X 771 (H)</td>
</tr>
<tr>
<td>Video Connection</td>
<td>BNC Female</td>
</tr>
<tr>
<td>S/N Ratio</td>
<td>48dB</td>
</tr>
<tr>
<td>Lens Type</td>
<td>5-50mm Zoom</td>
</tr>
<tr>
<td>Lens Control</td>
<td>Auto Iris DC Driven</td>
</tr>
<tr>
<td>Backlight</td>
<td>Built-in Backlight Compensation</td>
</tr>
<tr>
<td>Weight</td>
<td>27.87oz (790 grams)</td>
</tr>
<tr>
<td>(D) X (W) X (H)</td>
<td>7.64&quot; X 3.5&quot; X 7.52&quot; (19.4cm X 8.9cm X 19.1cm)</td>
</tr>
</tbody>
</table>
Table A.2: Specification of the Triplex Stand-Alone 4 CH DVR - CDR4060

<table>
<thead>
<tr>
<th>Specification</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TV System - NTSC</td>
<td>60 fields/sec</td>
</tr>
<tr>
<td>TV System – PAL</td>
<td>50 fields/sec</td>
</tr>
<tr>
<td>Resolution</td>
<td>NTSC 720x480</td>
</tr>
<tr>
<td>Video Inputs</td>
<td>4BNC, 1.0Vp-p Composite 75 ohms</td>
</tr>
<tr>
<td>Video Outputs</td>
<td>1.0Vp-p Composite 75ohms, 1 main monitor, 1 slave monitor</td>
</tr>
<tr>
<td>Real Time Display</td>
<td>120 frames/sec</td>
</tr>
<tr>
<td>Recording Speed</td>
<td>30 frames/sec, Programmable 1 to 30 frames/sec</td>
</tr>
<tr>
<td>Algorithm</td>
<td>MPEG</td>
</tr>
<tr>
<td>Recording Schedule</td>
<td>Continuous, Event Recording by Motion detection and Alarm sensor</td>
</tr>
<tr>
<td>Hard Disk Drive</td>
<td>80 GB</td>
</tr>
<tr>
<td>Replay</td>
<td>Forward, backward (fast, normal, stop, pause)</td>
</tr>
<tr>
<td>Alarm In</td>
<td>4 inputs</td>
</tr>
<tr>
<td>Alarm Output</td>
<td>1 relay output</td>
</tr>
<tr>
<td>Power consumption</td>
<td>Approx. 20W</td>
</tr>
<tr>
<td>Power Source</td>
<td>DC 12V 3A</td>
</tr>
<tr>
<td>Operating Temp.</td>
<td>41°F to 113°F</td>
</tr>
<tr>
<td>Humidity</td>
<td>10 to 90%</td>
</tr>
<tr>
<td>Dimensions</td>
<td>13.66(W) x 10.35(D) x 2.05(H)</td>
</tr>
<tr>
<td>Weight</td>
<td>6 lbs.</td>
</tr>
</tbody>
</table>

Specification of the Exide Deep Cycle Battery:

a) Nautilus Gold Marine/RV

b) 12 Volt

c) 500 CCA; 625 CA;

d) Length: 12"; Width: 6-13/16"; Height: 9 1/2"

e) Lead/antimony construction - the right kind for cycling

f) Enveloped polypropylene separators - for enhanced vibration resistance

g) Rigid polypropylene case and cover - for longer life in the maritime environment

h) MCA @ 32°F Fahrenheit – 625 and Reserve Capacity – 180

i) Amp Hour (20 Hour Rate) – 105
Digi XStream RF Modem Product Summary:

a) ISM 900 MHz or 2.4 GHz operating frequencies.

b) 100 mW (900 MHz) or 50 mW (2.4 GHz) power output (up to 20 mile range).

c) RS-232/RS-485 interfacing built-in.

d) Commerical (0 – 70° C) or Industrial (-40 – 85° C) temperature ratings.

e) FCC/IC/ETSI/CE approved.

f) Advanced networking & low-power modes supported.

Table A.3: Specification of Digi XStream-PKG-R 900 MHz RF Modem

<table>
<thead>
<tr>
<th>Long Range Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indoor/urban Range:</td>
</tr>
<tr>
<td>Outdoor line-of-sight Range:</td>
</tr>
<tr>
<td>w/ 2.1 dB dipole antenna</td>
</tr>
<tr>
<td>Outdoor line-of-sight Range:</td>
</tr>
<tr>
<td>w/ high-gain antenna</td>
</tr>
<tr>
<td>Receiver Sensitivity:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Power Requirements</th>
<th>XStream-PKG-R 900 MHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power Supply Voltage</td>
<td>7.5 ± 5 V</td>
</tr>
<tr>
<td>Receive Current</td>
<td>500 MHz: 70 mA</td>
</tr>
<tr>
<td></td>
<td>2.4 GHz: 50 mA</td>
</tr>
<tr>
<td>Transmit Current</td>
<td>900 MHz: 170 mA</td>
</tr>
<tr>
<td></td>
<td>2.4 GHz: 180 mA</td>
</tr>
<tr>
<td>Power Down Current</td>
<td>&lt;1 mA</td>
</tr>
</tbody>
</table>
### Table A.4: Specification of 4-element Yagi-antenna

<table>
<thead>
<tr>
<th>Specific Freq. (MHz)</th>
<th>865-980</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gain (dBi)</td>
<td>6</td>
</tr>
<tr>
<td>Maximum Power Input (Watts)</td>
<td>200</td>
</tr>
<tr>
<td>Product Narrative</td>
<td>4 element, 6 dB gain yagi, 1 piece unit construction with elements welded to boom. Feed system is enclosed in potted PVC radome for weatherability</td>
</tr>
<tr>
<td>Type</td>
<td>4-Element Yagi</td>
</tr>
<tr>
<td>General Freq. (MHz)</td>
<td>865-980</td>
</tr>
<tr>
<td>Bandwidth &amp; Rated VSWR (MHz)</td>
<td>84</td>
</tr>
<tr>
<td>Gain (dBi)</td>
<td>6.15</td>
</tr>
<tr>
<td>H. Beamwidth</td>
<td>106° Deg.</td>
</tr>
<tr>
<td>Vert. Beamwidth</td>
<td>85° Deg.</td>
</tr>
<tr>
<td>Front to Back Ratio (dB)</td>
<td>18 dB</td>
</tr>
<tr>
<td>VSWR</td>
<td>1.5:1</td>
</tr>
<tr>
<td>Polarization</td>
<td>Vert./Horiz.</td>
</tr>
<tr>
<td>Lightning Prot.</td>
<td>DC Ground</td>
</tr>
<tr>
<td>Size (HxWxD*)</td>
<td>12&quot;</td>
</tr>
<tr>
<td>Weight</td>
<td>0.68 lb</td>
</tr>
<tr>
<td>Rated Wind Velocity (MPH)</td>
<td>125</td>
</tr>
</tbody>
</table>

### Table A.5: Autoscope Rackvision specifications

<table>
<thead>
<tr>
<th>Communications</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>DB-9 RS-232 for local PC communication</td>
<td></td>
</tr>
<tr>
<td>One RJ-45 connector for RS-485 full-duplex network connection</td>
<td></td>
</tr>
<tr>
<td>Regulatory</td>
<td></td>
</tr>
<tr>
<td>NEMA T52 compliant</td>
<td></td>
</tr>
<tr>
<td>CE EN 55022</td>
<td></td>
</tr>
<tr>
<td>CE EN 61000-6-1</td>
<td></td>
</tr>
<tr>
<td>FCC Part 15, Class A</td>
<td></td>
</tr>
<tr>
<td>Power</td>
<td></td>
</tr>
<tr>
<td>12-24 VDC</td>
<td></td>
</tr>
<tr>
<td>&lt; 400 mA</td>
<td></td>
</tr>
<tr>
<td>&lt; 650 mA</td>
<td></td>
</tr>
<tr>
<td>AIS Cameras</td>
<td></td>
</tr>
<tr>
<td>22x continuous focus zoom lens</td>
<td></td>
</tr>
<tr>
<td>Horizontal: 5° to 73° FOV</td>
<td></td>
</tr>
<tr>
<td>Vertical: 5° to 58° FOV</td>
<td></td>
</tr>
<tr>
<td>Environmental</td>
<td></td>
</tr>
<tr>
<td>-29°F to -165°F (-34°C to -74°C)</td>
<td></td>
</tr>
<tr>
<td>0 to 95% relative humidity</td>
<td></td>
</tr>
<tr>
<td>Video Input</td>
<td></td>
</tr>
<tr>
<td>Composite 750 I Vpp, BNC connector</td>
<td></td>
</tr>
<tr>
<td>Color: NTSC or PAL</td>
<td></td>
</tr>
<tr>
<td>Black &amp; White, CCIR or RS-170</td>
<td></td>
</tr>
<tr>
<td>Output</td>
<td></td>
</tr>
<tr>
<td>1 Vpp, BNC connector</td>
<td></td>
</tr>
<tr>
<td>NTSC or PAL</td>
<td></td>
</tr>
<tr>
<td>Optional video compression</td>
<td></td>
</tr>
<tr>
<td>Dimensions and Weight</td>
<td></td>
</tr>
<tr>
<td>4.5&quot; H x 2.25&quot; W x 7&quot; L (114 mm x 57 mm x 173 mm)</td>
<td></td>
</tr>
<tr>
<td>0.9 lb (0.4 kg)</td>
<td></td>
</tr>
<tr>
<td>Warranty</td>
<td></td>
</tr>
<tr>
<td>Two-year warranty</td>
<td></td>
</tr>
<tr>
<td>Extended warranty available (five-year warranty package)</td>
<td></td>
</tr>
<tr>
<td>Product Support</td>
<td></td>
</tr>
<tr>
<td>Product support and training by team of factory-trained Autoscope technical support specialists</td>
<td></td>
</tr>
</tbody>
</table>
BIBLIOGRAPHY


[34] Katrin stark and Siegfried Fuchs, ”A method for tracking the pose of known 3-D objects based on an active contour model”, Proceedings of the 13th International Conference on Pattern Recognition, Volume1 , pp. 905-909, August 1996.


[64] Zeheang Sun1, George Bebis2 and Ronald Mille, "On-Road Vehicle Detection Using Optical Sensors: A Review"


[67] Luis A. Alexandre, Aurelio C. Campilho, "A 2D Image Motion Detection Method Using a Stationary Camera."


122

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.


[73] Hauppauge frame grabber http://hauppauge.com


[75] http://cs.gmu.edu

VITA
Graduate College
University of Nevada Las Vegas

Naveen Chintalacheruvu

Address:
4248 Grove Circle
Apt # 3
Las Vegas, NV-89119

Degree:
Bachelor of Technology, Electronics and Instrumentation Engineering, 2004

Thesis Title: Video based Vehicle Detection for Advance Warning Intelligent Transportation Systems

Thesis Examination Committee:
Chairperson, Dr. Venkatesan Muthukumar
Committee Member, Dr. Emma Regentova
Committee Member, Dr. Mei Yang
Graduate College Representative, Dr. Laxmi Gewali