

FHWA/IN/JTRP-2001/18

Final Report

**DEVELOPMENT OF A PORTABLE VIDEO
DETECTION SYSTEM FOR COUNTING TURNING
VEHICLES AT INTERSECTIONS**

**Andrzej Tarko
Robert Lyles Jr.**

January 2002

Final Report

FHWA/IN/JTRP-2001/18

**DEVELOPMENT OF A PORTABLE VIDEO DETECTION SYSTEM
FOR COUNTING TURNING VEHICLES AT INTERSECTIONS**

By

Prof. Andrzej P. Tarko
Associate Professor
Principal Investigator
and
Robert Scott Lyles Jr.
Research Assistant

School of Civil Engineering
Purdue University

Joint Transportation Research Program
Project No: C-36-17XX
File No: 8-4-50
SPR-2394

Conducted in Cooperation with the
Indiana Department of Transportation
and the
U.S. Department of Transportation
Federal Highway Administration

The contents of this report reflect the view of the authors, who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the Indiana Department of Transportation or the Federal Highway Administration at the time of publication. The report does not constitute a standard, specification, or regulation.

Purdue University
West Lafayette, IN 47907

January 2002

1. Report No. FHWA/IN/JTRP-2001/18	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle Development of a Portable Video Detection System for Counting Turning Vehicles at Intersections		5. Report Date January 2002	
		6. Performing Organization Code	
7. Author(s) Andrzej Tarko and Robert Lyles Jr.		8. Performing Organization Report No. FHWA/IN/JTRP-2001/18	
9. Performing Organization Name and Address Joint Transportation Research Program 1284 Civil Engineering Building Purdue University West Lafayette, Indiana 47907-1284		10. Work Unit No.	
		11. Contract or Grant No. SPR-2394	
12. Sponsoring Agency Name and Address Indiana Department of Transportation State Office Building 100 North Senate Avenue Indianapolis, IN 46204		13. Type of Report and Period Covered Final Report	
		14. Sponsoring Agency Code	
15. Supplementary Notes Prepared in cooperation with the Indiana Department of Transportation and Federal Highway Administration.			
16. Abstract <p>This research was conducted to test the feasibility of using existing video-detection techniques for counting turning volumes with a portable installation. This was accomplished by integrating a forty-five foot mechanical tower mounted on a van with two video detection systems, Autoscope and VideoTrak.</p> <p>The research project has produced results in three categories: prototype methods of counting turning volumes, evaluation results, and general specifications of a portable video-based system for counting vehicles at intersections. The method based on spot detection uses redundancy of data (more spots than movements) to improve the results quality. The method for VideoTrak one-dimensional tracking classifies maneuvers based on the location where vehicles enter and exit a tracking strip. Both the evaluated methods in their current versions do not meet the accuracy expectations expressed by the INDOT representatives.</p> <p>The general system specifications were developed to help build a prototype unit. The biggest challenge is the structure of the system that has to be portable, stable during data collection, and protected against tempering with. The authors advise postponing building a prototype system by the time needed to develop satisfactory image processing and interpretation software for identifying vehicles' maneuvers at intersections.</p>			
17. Key Words video detection, counting vehicles, turning movement, Autoscope, VideoTrak.		18. Distribution Statement No restrictions. This document is available to the public through the National Technical Information Service, Springfield, VA 22161	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 162	22. Price

TABLE OF CONTENTS

	<u>Page</u>
LIST OF TABLES	v
LIST OF FIGURES.....	vi
NOMENCLATURE.....	ix
1. IMPLEMENTATION REPORT.....	1
2. INTRODUCTION.....	3
3. LITERATURE REVIEW.....	6
3.1 Video Image Processing Systems (VIPS).....	6
3.2 Turning Volume Estimation.....	9
4. RESEARCH OBJECTIVES AND METHODOLOGY.....	12
5. MOBILE TRAFFIC LABORATORY.....	14
5.1 Introduction.....	14
5.2 Selecting Video Detection Systems.....	15
5.2.1 Autoscope (Standard) Description.....	16
5.2.2 VideoTrak Description.....	18
5.3 Integrating Video Detection Systems.....	20
5.4 Detail Design.....	21
5.5 Van Integration.....	25

	<u>Page</u>
6. DATA COLLECTION.....	34
6.1 Potential Factors of Video System Performance.....	34
6.2 Data Collection Plan.....	39
6.3 Field Data Collection	42
6.4 Ground Truth.....	45
7. AUTOSCOPE METHOD AND EVALUATION.....	49
7.1 Detector Matrix Method.....	50
7.2 Flow Conservation Method.....	55
7.2.1 Estimation Method	58
7.2.2 Simulation Test	60
7.2.3 Preliminary Regression Analysis	62
7.3 Data Extraction.....	64
7.4 Method Evaluation	67
7.4.1 Autoscope Descriptive Statistics Results	71
7.4.2 Linear Regression Results.....	75
7.5 Closure	78
8. VIDEOTRAK METHOD AND EVALUATION.....	81
8.1 Academia Version of VideoTrak	82
8.2 Tracking Strip Method	85
8.3 Tracking Strip Algorithm.....	88
8.4 Data Extraction.....	93

	<u>Page</u>
8.5 Method Evaluation	94
9. COMPARATIVE EVALUATION.....	99
9.1 Implementation Issues.....	99
9.2 Performance Measures	100
9.3 Strengths and Weaknesses	102
10. GENERAL SYSTEM SPECIFICATIONS (DRAFT).....	104
10.1 System Design and Operation	105
10.1.1 Exterior Components	105
10.1.1.1 System Structure	106
10.1.1.2 Data Acquisition System.....	106
10.1.1.3 Data Storage and Processor System.....	107
10.1.1.4 Power Supply System	107
10.1.2 Interior Components.....	108
10.1.2.1 Data Storage and Processor System.....	109
10.1.2.2 Video Detection System.....	109
10.1.3 System Integration.....	110
10.1.4 System Operations.....	112
10.1.4.1 Offline Processing.....	112
10.1.4.2 Field Processing	115
10.2 Detailed Specification and Guidelines	117
10.2.1 System Structure	117
10.2.2 Data Acquisition System.....	124

	<u>Page</u>
10.2.3 Data Storage System	127
10.2.4 Power Supply	129
10.2.5 Video Detection System.....	131
11. CLOSURE.....	134
LIST OF REFERENCES	137
APPENDIX	139

LIST OF TABLES

<u>Table</u>	<u>Page</u>
6.1 Selected intersections	40
7.1 Numbering of the detectors and flows at a four-leg intersection	56
7.2 Effect of incorrect weights on the regression results ($D^{0.5}$ is correct).....	61
7.3 Example of Extracted Autoscope Data	67
7.4 Descriptive Statistics for Entire Data set (Mobile Laboratory and PTZ).....	72
7.5 Descriptive Statistics for Data collected with Mobile Laboratory	74
7.6 Descriptive Statistics for Data collected with PTZ	75
7.7 Regression results for Equation 6.10.....	77
7.8 Regression results for Equation 6.11.....	78
8.1 Results of VideoTrak detection system.....	95
10.1 Honda 3000is Specifications.....	130

LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
5.1 Example Autoscope Detector Layout	17
5.2 Example VideoTrak field-of-view Layout.....	20
5.3 Schematic Layout of Video Detection System	23
5.4 Example Mast Setup.....	23
5.5 Camera Mounting System.....	24
5.6 Guide wire spool attached to concrete block	24
5.7 Control Station Layout (Facing Right Side of Van)	28
5.8 Van Schematic Layout	29
5.9 Portable Video Detection Control Station.....	31
5.10 Floor Layout of Portable Video Detection System	32
6.1 Camera Offset	36
6.2 Data Inventory Worksheet	40
6.3 Intersection Illustrations.....	41-42
6.4 Intersection Data Collection W.S.....	44
6.5 Intersection Condition Observation Data Sheet.....	45
6.6 Example data format for Analysis.....	48
7.1 Example Detector Matrix Layout.....	51
7.2 Example of Requirement 1.....	53

<u>Figure</u>	<u>Page</u>
7.3 Example of Requirement 2.....	53
7.4 Example Intersection with detector locations	56
7.5 Detector-flow assignment matrix for the example intersection	57
7.6 Actual detector counts vs. ideal detector counts	63
7.7 Turning Flow estimates vs. ground truth flows	64
7.8 Examples of detector layouts	66
7.9 Estimated Flow vs. Ground Truth.....	73
8.1 Vehicle traveling through tracking strip.....	85
8.2 Example of tracking strip method	87
8.3 Reported Start and End pixels for VideoTrak.....	91
8.4 Algorithm for Tracking Method Program.....	92
8.5 Example of Tracking Strip Layout.....	94
9.1 Comparative Evaluations of Autoscope and VideoTrak.....	101
10.1 Fundamental design.....	110
10.2 Exterior Component Connectivity	111
10.3 Interior Component Connectivity	112
10.4 Offline Processing Operations	114
10.5 Field Processing Operations.....	116
10.6 Model 802 Triler-mounted masts.....	120
10.7 Model 802 Dimensions	120
10.8 Model 802 Specifications.....	121
10.9 Model 802 Example Setup	122

<u>Figure</u>	<u>Page</u>
10.10 Floatograph Trailer-mounted mast.....	123
10.11 Floatograph Trailer-mounted mast setup	123
10.12 Panasonic data acquisition equipment.....	126
10.13 Panasonic parts.....	127
10.14 Honda 3000is Power Generator	129
10.15 Vantage Plus by Odectics.....	132
10.16 Autoscope 2004LE by Econolite Control Products, Inc.	132
10.17 Autoscope 2004LE by Econolite Control Products, Inc.	133

NOMENCLATURE

Symbols

CE	Counting Error;
CE	Absolute Value of Counting Error;
NO	Number of Observations;
MC	Mean Count;
ME	Mean Error;
SE	Standard Error;
RME	Relative Mean Error;
RSE	Relative Standard Error;
Hei	Indicator variable for height of camera;
Cam	Indicator variable for number of cameras;
Lmid	Indicator for midday light conditions;
Leve	Indicator for evening light conditions;
Wmod	Indicator for moderate wind conditions;
Wheav	Indicator for heavy wind conditions;
Twsc	Indicator for two-way stop control;
Sig	Indicator for two-way stop control;
Le	Indicator for left turn movement;
Ri	Indicator for right turn movement;

La	Number of lanes of the intersection minus four;
Sing	Number of single unit trucks;
Truck	Number of semi-trailer trucks;
Ped	Number of pedestrians;
Ni	Indicator for night conditions;
Rain	Indicator for rain conditions;
Snow	Indicator snow conditions.

1. IMPLEMENTATION REPORT

The research project has produced results that can be summarized in three categories:

- (1) Two distinct prototype methods of counting turning volumes, one for the spot detection techniques such as Autoscope, and one for a one-dimension vehicle tracking used in the VideoTrak system,
- (2) Evaluation results of the two mentioned systems used for counting turning volumes at selected intersections,
- (3) General specifications of a portable video-based system for counting vehicles at intersections.

The method based on spot detection can be easily implemented even through a programmed Excel spreadsheet. The method uses the standard features of the Autoscope system. The method based on the VideoTrak one-dimensional tracking requires a special format of data produced by, so called, Academia version. Current limitations of the VideoTrak software make the method difficult to implement due to the extensive post-processing time.

The implementation is envisioned in two steps: (1) Building and testing a prototype unit, (2) Full-scale implementation of the modified unit. The general system specifications were developed to help build a prototype unit. The specifications include example components found on the market today. The biggest challenge is the structure of the

system that has to be portable, stable during data collection, and protected against tempering with. The cost of a complete prototype system is estimated to range between \$80,000 - \$110,000 according to 2001 prices. The final cost depends on the system configuration.

The authors advise postponing building a prototype system by the time needed to develop satisfactory image processing and interpretation software for identifying vehicles' maneuvers at intersections. The Purdue team will build a portable system (mobile traffic lab), which will meet the developed general specifications for the video acquisition system and for the data storage/processing component. The system will serve two purposes: (1) To test the system abilities to acquire and store high quality video from two channels in a sustain manner for an extended period. (2) To create a testing facility for a new generation of vehicle-tracking algorithms.

A prototype system is proposed to be built by a selected contactor and according to the current specifications with possible future modifications after a positive test of counting accuracies and equipment reliability are obtained.

2. INTRODUCTION

Intersection traffic data including classification counts are primary inputs to many transportation studies, analyses, and designs. Currently, there are three common methods of obtaining these counts; using an automatic traffic recorder, portable machine traffic recorder, or to perform manual classification counts. Automatic traffic recorder counts are permanent installations used for continuous counting, and are the most expensive considering equipment costs along with pavement damage (i.e. embedded loop detectors). The portable machine traffic recorders are used for shorter counting periods, and they are relatively inexpensive. Pneumatic tubes and non-intrusive radar are the most common types used, but are found to be less reliable. More than often manual counting is performed during intersection data collection. Manual counting occurs when one or more observers' counts traffic for an extended period of eight to sixteen hours depending on the traffic study. This technique is both labor intensive and expensive, and may not be accurate if the traffic intensity exceeds observers' counting capabilities. Therefore, a method needs to be developed that will improve the cost effectiveness and accuracy of traffic counting at intersections.

Recent technological advancements in image processing have resulted in several video detection systems. Although, these systems are designed primarily for traffic control, they are capable of performing non-intrusive collection of traffic characteristics such as vehicle counts, speeds, and classification. Past research of particular video

detection systems have an overall good rating and are currently in use in many agencies across the U.S. In addition, video detection systems have the potential to perform multiple detections' along with a noticeable flexibility in spot selection.

Several years ago, the Indiana Department of Transportation (INDOT) initiated use of video detection for counting flows at intersections. Preliminary benefit-cost estimations indicated that a portable system settable in short time with unattended operation for extended period can bring savings on labor costs that exceed the system's purchase, operations, and maintenance costs. The INDOT Greenfield district was testing a portable system that included a van equipped with three cameras, a thirty foot telescoping pneumatic mast, and the Autoscope video detection system. The resulting counting errors were exceedingly considerable. In addition, turning flows could not be measured at intersections where turning movements did not use exclusive approach lanes. These problems were attributed to the effect of occlusion in an image, the multiple detection of extended vehicles, and the incapability of the Autoscope video detection system to track vehicles.

The objective of this research was to provide specifications for a functioning portable video detection system that can eliminate or at least mitigate the drawbacks of the system being used by INDOT Greenfield district. This was accomplished by integrating a forty-five foot mechanical tower mounted on a van with two video detection systems, Autoscope and VideoTrak. In addition, there was an attempt to enhance the Autoscope system and utilize VideoTraks' capability of tracking vehicles to obtain and classify turning volumes. Videotaped traffic data was collected for several intersections, and a comparative evaluation of both video detection systems was completed to prepare final

specifications for a functional design. The subsequent chapters explain the methodologies used in this research, and discussions related to the analysis of results obtained, along with final specifications and guidelines for a functioning system.

3. LITERATURE REVIEW

Video detection is relatively new to the transportation industry and was developed through Intelligent Transportation System (ITS) technology. Although there are many commercial systems available today, most agencies around the country are still evaluating these systems to investigate their usefulness. At this time, there has been no research done to utilize video detection to count all turning movements at intersections. However, there have been some computational methods developed in the past that use statistical and probability techniques together with known approach counts. This chapter will give explanation to the current status of video detection, and methods involved to obtain turning movement counts.

3.1 Video Image Processing Systems (VIPS)

Video image processing systems can be used to analyze video data collected with Closed Circuit Television (CCTV) systems. Machine vision technology combines video imaging with computerized pattern recognition. Recent technological advancements along with reduced computer and image processing hardware costs have made VIP detection systems an attractive and viable alternative for collecting traffic data.

The advantage of VIP detection over traditional surveillance means lies in its area detection capabilities within a camera's field of view. This allows the detection of spatial traffic parameters, such as density, queue lengths, and speed profiles, which usually cannot be easily obtained by conventional methods. In addition, video detection is able to provide additional information such as traffic on the shoulders, stopped vehicles, lane changing, speed differential, and traffic slow downs in the other direction. VIPS generally fall into the two categories: tripwire systems and tracking systems.

The majority of the commercial VIPS available today are tripwire systems. These systems operate with the use of virtual detectors that imitate the operation of loop detectors, but they do not track vehicles. Rather, they are capable of identifying individual vehicles and follow their movements in time. The following are examples of commercial tripwire systems: AUTOSCOPE, CCATS, TAS, IMPACTS, and TrafficCam (Coifman, 1998). The systems typically allow the user to specify several detection zones in the video image, and then the given video detection system recognizes the changes in image intensity to indicate vehicle presence/passage. The primary advantages of these systems are the ease of placing detector zones, the fact that there is no need to cut into pavement, and that some of these systems are capable of utilizing a large number of detection zones.

There are some commercial video detection systems that do track vehicles. Examples include the CMS Mobilizer, Eliop EVA, PEEK VideoTrak, Nestor TrafficVision, and Sumitomo IDET (Coifman, 1998). Generally these systems use region based tracking, where the entire video image is scanned for pixel changes, looking for a vehicle to follow along the roadway. The advantage of vehicle tracking is that even with

a moving background the detection algorithm should pick out a true vehicle.

Additionally, vehicle tracking can determine when a vehicle is changing lanes. In both methodologies, a near vehicle in the camera field of view can occlude a far vehicle, but vehicle tracking lessens the effect of vehicle occlusion.

Over the past several years, there has been much research into the evaluation of video image processing systems. In 1994, the Virginia Department of Transportation (VDOT) along with the Maryland State Highway Administration (MSHA) conducted one of the earliest VIPS evaluations (Cottrell, 1994). The purpose of the study was to evaluate the capabilities of the Autoscope video detection system for incident management to combat urban freeway congestion. Although the objective to assess the performance of the VIDS for incident detection was not accomplished, an examination of its capability to monitor traffic was achieved. In general, they found that speed and volume measurements were inconsistent, and that volumes detected were significantly greater than the volumes measured by loop detectors. Also, they found that camera placement above the travel lanes yields better results than cameras placed at the side of the road.

A more recent study occurred in 1998 by the Minnesota Department of Transportation (MnDOT) and SRF Consulting Group, Inc. (Bahler, et al, 1998). It included a field test of nonintrusive traffic detection technologies, including the following video detection systems: Trafficam S (Rockwell International), Autoscope 2004 (Image Sensing Systems), EVA 2000 S (Eliop Trafico S.A.), and VideoTrak 900 (Peek Transyt). The devices were tested in a variety of environmental and traffic conditions at both intersection and freeway test sites. They found that video devices are not well suited for temporary counting since video requires extensive installation and set-up time. Also,

video detection seemed to perform erratically, especially at the intersection test site because of congested stop-and-go traffic. Weather and other environmental variables were found to have minimal impact, but lighting conditions, wind, and snow had a significant impact on the video detection.

Last year the Texas Transportation Institute in cooperation with the Texas Department of Transportation further evaluated the VideoTrak 900 system upon findings from the MnDOT (Middleton, 2000). Testing of the video detection system occurred on a freeway test bed with low to moderate free-flow traffic. The parameters measured for accuracy were vehicle presence and speed, along with installation cost, ease of setup, and calibration. It was found that the Peek VideoTrak system's presence and speed accuracy both declined to unacceptable levels during nighttime and during rain. It was also the most difficult to set up and the most expensive.

In summary, evaluations of commercial VIPS find that the systems have problems with congestion, high flow, occlusion, camera vibration due to wind, lighting transitions between night/day and day/night, and long shadows linking vehicles together (Coifman, 1998). Excluding these types of environmental conditions, VIPS have reasonably good detection capabilities.

3.2 Turning Volume Estimation

An investigation of past research found several papers regarding the measurement of turning flows at intersections. All of which can be described with the presentation of

two distinct approaches. The first approach proposes to extract turning volumes from vehicle spot counts, whereas the second suggests identifying individual vehicle maneuvers to categorize turning movements.

This first approach includes methods of estimation when the number of turning volumes is greater than spot detection counts. Typically, eight detectors are used to gather data on four entrances and exits of a four-leg intersection. The contrasting methods describe possible sources of information needed to complete the data set. Hauer et al (1981) proposes the use of the maximum likelihood assumption to estimate the turning flows. Van Zuylen (1979) and Mountain and Westwell (1983) suggested the use of either observed estimated turning proportions. Ploss and Keller (1986) applied the entropy assumption to the known information of travel times between detectors. The travel times were used to improve temporal traffic consistencies across a series of detectors. Cremer and Keller (1987) extended Ploss and Keller's concept to the use of entrance and exit detectors at an intersection. The accuracy of these methods relies heavily upon the quality of the spot counts along with the validity of the assumptions.

The second approach includes methods that attempt to identify vehicle maneuvers during individual vehicle detection. Lu et al (1988) introduced the method of an automated recognition of turning signals for identifying vehicle maneuvers. Unfortunately, the concept provided many problems, and was unsuccessful. Most recently, Virkler and Kumar (1988) presented a method of using multiple detectors strategically placed at intersection corners to identify turning maneuvers. However, this method can only be applied to signalized intersections where information about signal states are known. Another possibility is to employ video detection systems that possess

tracking capabilities. Unfortunately, there is no sufficiently reliable tracking system available today.

Consequently, no agency has been able to implement those methods into their operations.

Therefore, a method needs to be developed to replace the impracticality of the existing manual counting methods. Recent advances in spot detection using vision technology encourage revisiting the first approach to estimate turning volumes at an intersection.

Present video detection systems have a short setup time along with the ability to place large number of reliable detectors on a video image. The concept is to use the assumption of flow conservation to estimate turning volumes from multiple flow counts, and then utilize data redundancy to improve the estimation accuracy.

4. RESEARCH OBJECTIVES AND METHODOLOGY

The primary objective of this research is to assess whether current video image technologies allow for designing a portable system for accurately counting turning vehicles at intersections. The additional objective is to develop draft specifications to ascertain the feasibility for developing such a prototype system. The successful results from this study will provide innovative methods for utilizing standard features of video detection systems to count turning vehicles. Furthermore, the findings will assist transportation agencies in the decision to further advance the concept of a portable video detection system.

To collect data, an empirical study was performed. To evaluate selected video detection systems accuracy in estimating turning volumes at intersections, a portable installation device was substituted for an actual portable video detection. The mobile traffic laboratory, which consisted of a vehicle and an attached camera mast, was used as a tool to imitate the necessary function of a portable video detection system under evaluation. Furthermore, an existing mounted camera was used to collect data during inclement weather conditions, when the portable installation device was incapable of doing so.

The performance of the video detection systems was measured with a counting error. The counting error was the difference between the count estimate and the ground truth count. The ground truth data was obtained by direct observation of video images,

and not using other detection methods, such as inductive loops. To provide confidence of the evaluation, the taped video images were visually examined to extract and document the relevant traffic and environmental characteristics. Given that the ground truth data was acquired from videotapes, all human counting errors were alleviated with multiple playbacks of the tapes. Descriptive statistics described the aggregated counting error. These include absolute and relative mean and standard errors. In addition, linear regression models were developed to investigate how local conditions and weather affect performance of video detection.

Ultimately, specifications for a functional design of a prototype was developed to further study the feasibility of the concept. An investigation of current available mast structures was performed to specify a superior unit for the prototype. Moreover, there was research into the types of video data storage available to record a large amount of data. In conclusion, an analysis was done to show the feasibility of a portable video detection system.

5. MOBILE TRAFFIC LABORATORY

A mobile traffic laboratory was used as a portable installation to evaluate the video detection systems. Prior to its operation, two detection systems had to be selected from all the available systems, and then integrated with the existing lab components. This chapter explains the partial development and integration of a mobile traffic laboratory used in the research. First, the selection process to identify two video detection systems for the mobile lab is described. Then the initial interior of the mobile lab is explained, followed by a detailed description of the integrated mobile traffic laboratory

5.1 Introduction

One of the concepts of this research was to test the feasibility of providing portability to video detection, and understand how portability can affect the reliability of traffic data obtained from video detection equipment. Therefore, a portable unit was assembled to create circumstances when a genuine portable video detection system is in use. This portable unit incorporates all necessary components required for a fully functioning portable video detection system.

5.2 Selecting Video Detection Systems

The video detection system component is an integral feature of a portable video detection system. Since the system must be economical and must possess required features, an extensive search of current video detection products was conducted. The comprehensive search revealed six potential products available on the market. These are listed below with their corresponding manufacturers:

- VideoTrak 905 – Peek Traffic Systems
- TrafficVision – Nestor Traffic System, Inc.
- Autoscope 2004 Standard– Autoscope, Inc.
- Moniwatch – Monitron
- CAMDAS (Camera Data Acquisition System) – ARRB Transportation Research
- Traficon – Control Technologies

An assessment of the products and companies was completed to select the most promising video detection systems for use in this research. Autoscope 2004 (Autoscope, 2001) was selected because it is the video detection technology that INDOT endorsed for this research. The CAMDAS and Traficon products are sold in Australia and Belgium, respectively. Though the products had a potential use for the portable video detection system, the locations of their manufacturers could have posed a potential communications problem in research and implementation. The Moniwatch software is a similar technology to Autoscope. It does not possess vehicle-tracking capabilities, and therefore was eliminated. Finally, the products offered by the Peek Traffic Systems and Nestor Co.

was found to be promising for this research. They were both offered in the U.S. and claimed to have tracking capabilities. Nestor Traffic Systems Inc is a relatively new company that introduced their product in 1997. Conversely, Peek Traffic System is an established company that provides products for the traffic control industry. For that reason, VideoTrak (Videotrak, 2001) was selected as the second video detection system for evaluation.

5.2.1 Autoscope (Standard) Description

The Autoscope 2004 detection system is considered a first-generation (tripwire) video detection unit made up of a video image-processing (VPU) unit along with a video-graphics card and user-friendly software. The Autoscope Machine Vision Processor (MVP) unit is a box that contains a microprocessor-based CPU, specialized image processing boards, and software to analyze video images. The MVP accepts up to four video inputs from multiple image sensors to provide wide area vehicle detection for traffic parameter extraction.

Using a mouse and interactive graphics, the user sets up an Autoscope detector layout by placing "virtual detectors" on the video image displayed on a monitor (Figure 5.1). Each detector represents a zone, either a wide area zone or a short zone, which in the simplest form emulates an inductive loop. The virtual detectors are grouped pixels on the screen that can either take the form of a square or rectangle, and their size and shape can be manipulated. There is a limit of ninety-nine detection zones that can be assigned to the Autoscope processor. Information from various detection zones can also be combined into the following logical operations: AND, OR, NAND, and N of M.

Once the system is set up and operating, a detection signal is generated each time an object crosses a virtual detector. The virtual detector changes color when a detection signal is generated, therefore, the user can easily check whether the virtual detector is properly working. Ultimately, the Autoscope processor generates traffic data including volume, speed, occupancy, headways, queue lengths, and vehicle classification. These traffic data can be collected in two ways, either as interval data or as event data. Interval data is gathered in consecutive time intervals specified by the user. Event data is gathered for every object that is detected. This is very synonymous to per vehicle record data collected from loop detectors.

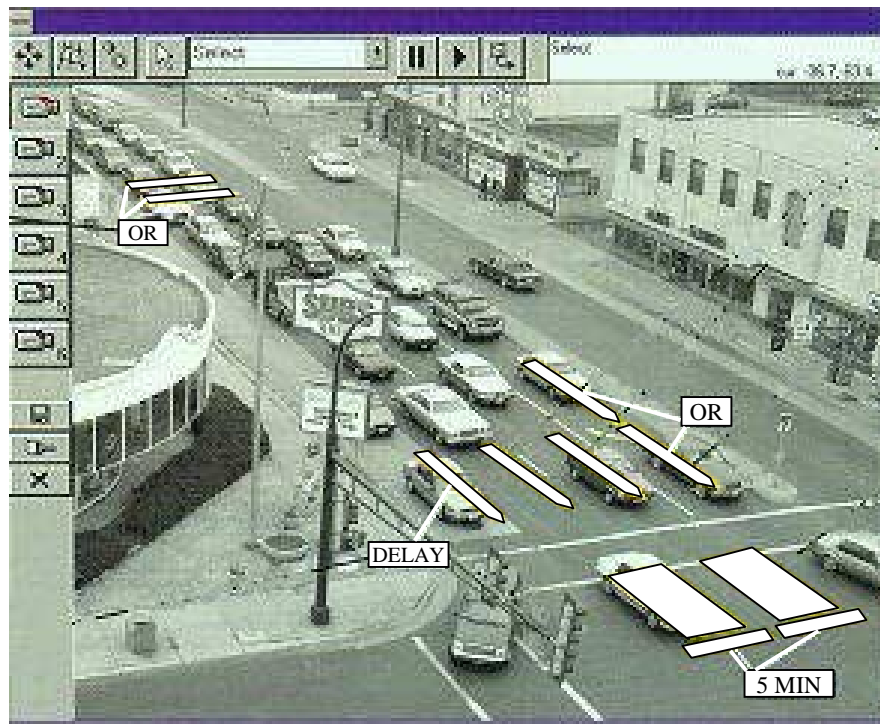


Figure 5.1 Example Autoscope Detector Layout

5.2.2 VideoTrak Description

The VideoTrak 905 is considered a second-generation (tracking) video detection system. Similar to Autoscope, it is comprised of a video tracking unit (VTU), microprocessor-based CPU, specialized image processing boards, and software to analyze video images. Instead of using a separate video card, VideoTrak digitizes the analog video signal within the VTU. The VTU can accept up to five video inputs from multiple image sensors and one video output. VideoTrak tracking algorithms minimizes vehicle misses and false detection common in previous tripwire detection systems by blobifying pixels to represent moving vehicles. Then the system can determine the location of the vehicles from pixel intensity changes that occur from frame to frame in a video image.

Using a mouse and interactive graphics, the user sets up a VideoTrak field of view layout by placing virtual detectors and tracking strips on the video image displayed on a monitor (Figure 5.2). A tracking strip represents an area where tracking takes place. A detector represents a zone placed within the tracking strip to gather data. There are up to 32 possible detection zones per camera field-of-view that can be drawn using 2, 3, or 4 points. The detection zones can overlap and intersect and span multiple tracking strips. Similar to Autoscope, information from various detection zones can be combined into logical operations (AND, OR, NAND, and N of M).

Tracking strips may be of various size and orientation and are typically associated with a lane, shoulder, or other areas of interest. Tracking strips are polygons comprised of

four to eight points, and tracking is based on one dimensional flow and direction (vertical or horizontal). An algorithm tracks vehicles by initially obtaining average pixel values for either rows or columns within the strip, and then recognizing a change in those average pixel values.

Once the system is set up, a detection signal is generated each time a vehicle crosses a virtual detector. Similar to Autoscope, VideoTrak visualizes vehicle detections with changes in detector color. Ultimately, the VideoTrak processor generates traffic data including volume, speed, occupancy, headways, queue lengths, vehicle classification, and delay.

Traffic data for a field-of-view is stored every twenty-four hours within the VTU (12 a.m. to 12 p.m.). Hence, data can only be recovered from the previous day or days. Traffic data is retrieved using a separate 'Report Utilities' software program. It is recovered by first declaring which day one would investigate. Then through a step-by-step procedure, relevant information such as which detector and traffic data types, can be acquired in selectable time intervals of 10, 20, or 30 seconds; or 1, 5, 10, 15, 30, or 60 minutes. Since VideoTraks' standard features do not provide per-vehicle-record data for each detector, the Academia version software was obtained.

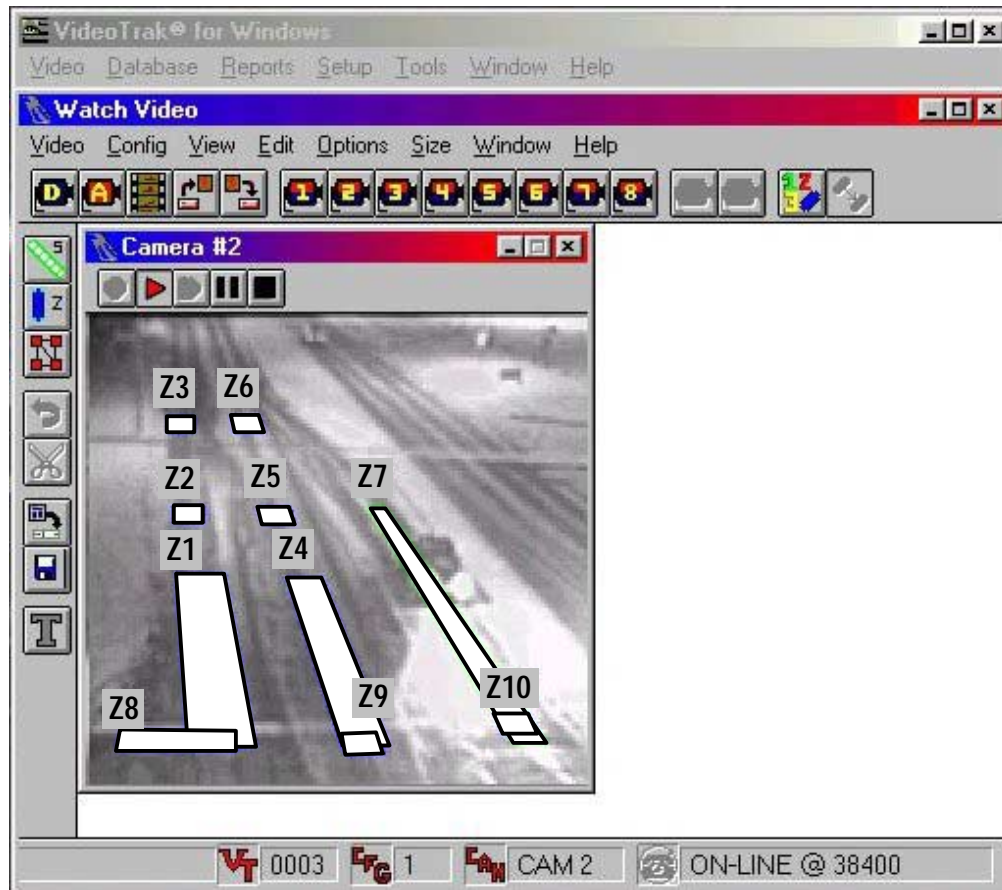


Figure 5.2 Example VideoTrak field-of-view Layout

5.3 Integrating Video Detection Systems

The Joint Transportation Research Project at Purdue University uses a utility vehicle for data collection of all regions in Indiana. A short while ago this vehicle was replaced, and the old utility vehicle was structurally and mechanically prepared for research purposes. This vehicle is a 1988 Ford Van that was equipped with only safety features needed for data collection. These features include strobe lights on each side, caution lights on the back window, and a citizens band (CB) radio for emergency

communication. Afterwards, the van was rigged with a telescoping mast with camera mount that could be used toward future research projects involving video detection.

For this research the inside of the van was constructed to house all the equipment needed for both video detection systems. Figure 5.3 is an illustration of how the electrical components of the portable video detection are connected, and the following sections describe in detail how the van was constructed.

5.4 Detail Design

As previously stated, the research van had already possessed a mechanical mast with detachable camera mount. The mast is a product of Floatograph Technologies located in Napa, California. It was designed specifically for this van's dimensions. When not in use, the mast rests on two steel crossbars that are fixed to the roof of the van not to cause damage to the roof. Directly at the back of the van, along the length of the mast, there is a pivot point where a steel bar joins the mast to the hitch of the van. The inverted 'T' shaped steel bar has attached legs for reinforcement and stabilization of the mast to the ground. When the function of the mast is utilized it is hoisted from its pivot point to an initial vertical position where it can then be raised to a specified height (Figure 5.4).

The mast is constructed of four independent sections of weatherproof aluminum designed with a square cross-section. These sections use a four-pulley system with a supporting 1/8" stainless steel cable for vertical extension. The bottom section of the mast has affixed an electric winch that is joined to the pulley system by the steel cable. The

electric winch operates the vertical extension of the mast, and is powered by a 12 V car battery. The bottom section of the mast also has affixed two leveling units to ensure horizontal and vertical equilibrium of the mast.

The top end of the mast allows for the attachment of a heavy-duty camera system (Figure 5.5). This camera system consists of two pan and tilt heads on a camera mount, weather resistant camera covers, a control cable, and a control console. The pan tilt heads provide for a 360° pan and 90° tilt of two camcorder cameras attached to the camera mount. The camera covers protect the camcorders from harsh weather elements including heat generated from direct sun exposure, and the accumulation of moisture from either rain or snow conditions. The 80-foot control cable is a composite of electrical wires for the power, video, audio, and connects to the control console. The control console has a 4" color monitor that fastens to the control console. The control console provides power and remote control (pan/tilt) to the pan/tilt mechanism and video cameras. Figure 5.3 illustrates how the camera system functions.

Previous research with the telescoping mast discovered that it must be stabilized in high wind situations. Therefore, researchers designed a method of using guide wires to secure the mast in vertical extension. The individual guide wires connect to a weighty concrete block by means of a metal spool with attached gear (Figure 5.6). This provides a method of modifying guide wire length along with its corresponding mast height, and applying an applicable tension needed to stabilize the mast.

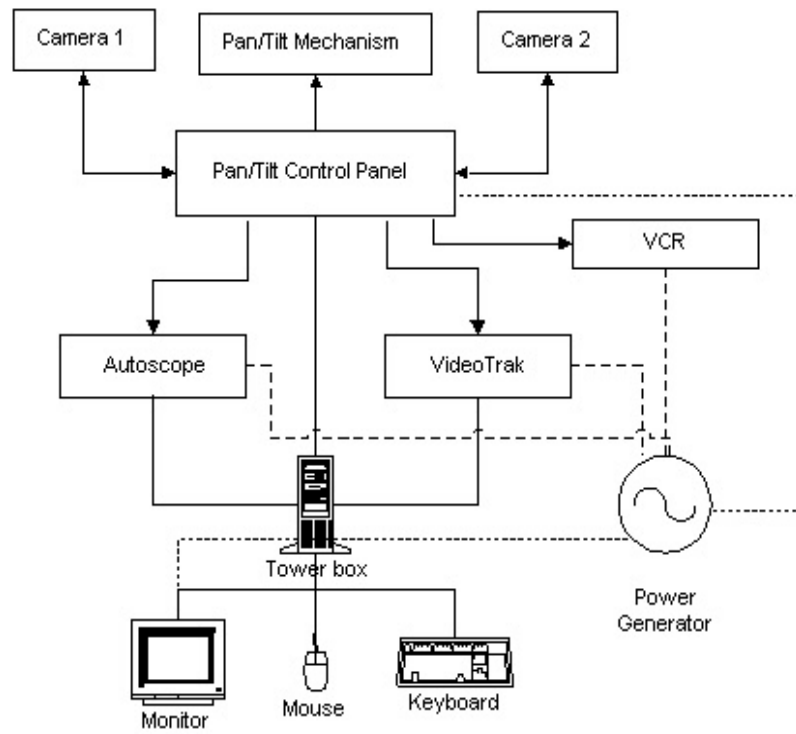


Figure 5.3 Schematic Layout of Video Detection System

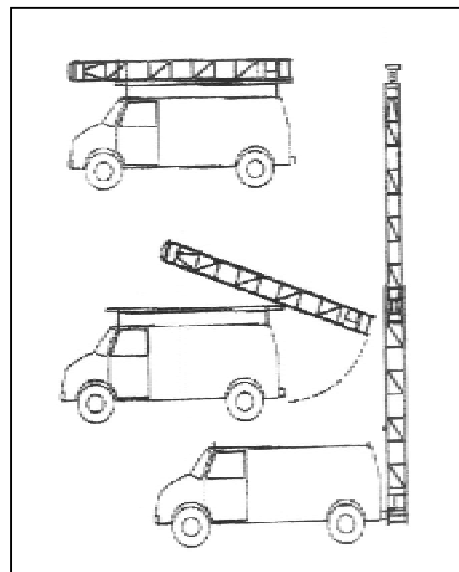


Figure 5.4 Example Mast Setup



Figure 5.5 Camera Mounting System



Figure 5.6 Guide wire spool attached to concrete block

As stated earlier, the addition of the telescoping mast to the van provided the means of developing future research with video detection technology. The concept of a portable video detection system having the capability of collecting traffic data at intersections utilizes the van with its mast. However, there had to be an integration of other sophisticated system components that can allow data to be collected onsite. These components include the necessary devices for video detection, an organized allocated space for the safe movement of video detection devices, and a proficient power supply.

5.5 Van Integration

The concept for the portable detection system allows for any user to be able to collect traffic data at intersections using video detection technology. The portability of the concept utilizes the van with equipped telescoping mast. Hence, there was a design developed to integrate all necessary devices and supplies into the existing interior of the equipped van for a complete system.

Foremost, a list was created to specify components needed to integrate video technology into the van. The list emanated from the necessary parts needed for video detection, which includes the visual processing unit that analyzes images, and a computer that stores the data and provides a graphical interface for developing detector layouts. Econolite's Autoscope 2004 and Peek Systems VideoTrak 905 were the two visual processing units chosen to evaluate the portable video detection system. The computer used in conjunction with the video detection systems was a 486 processor with a 15"

monitor, keyboard, and mouse. The list expanded by including a VCR that could be used to produce a VHS tape library of the data collected for the research, a gas generator that provides a proficient power supply, and the initial control console used for operating the camera system on top of the mast. A concept was developed to combine all of these major components into one control station inside the van where all operations of the system could be accomplished. The control station would be built as a racking system with shelving for the major components, and a chair for user operation. Other supplies needed included space for the concrete stabilizer blocks, the control cable, 12 V battery for the electrical winch, and other supplies used to set up the mast. These include pieces of wood, a ladder, and hydraulic jack. Below is the complete list of components used to design the interior of the van.

- Racking System with Shelves
- Chair
- Two Visual Processing Units (Autoscope, VideoTrak 905)
- Computer (Tower, Monitor, Keyboard, and Mouse)
- Control Console with 4 inch monitor
- 12 V Power supply box for Control Console
- VCR
- Gas Generator
- 4- Concrete Stabilization Blocks
- Hydraulic Jack Kit
- Control Cable

- Wooden planks
- Wooden ladder
- 12 V Battery
- Traffic Cones
- Camera mount with 2 camcorders
- Control Station

The concept of the control station considers the electrical components needed for video detection. These include the two video detection systems, computer equipment, control console equipment, and space for an alternative source of power. Foremost, a design layout of the station was developed (Figure 5.7). The station was designed to consider the configuration of the electronics between the hardware, and the accessibility of equipment needed to manage the control station.

The computer monitor along with the keyboard is placed directly in the middle of the unit. This was done for the convenience of the operator. The computer tower was located on the left side between the backdoor of the van and the unit. This was done because when the mast is hoisted there is no access to the van from the rear, and it is safe during high periods of acceleration and deceleration of the van. Both of the video detection systems are placed left of the computer terminal, since they only need to be disturbed during un-installation. On the right side of the monitor are the VCR and control console. The VCR was located above the control panel. The control station layout was designed to

consider the configuration of the electronics between the hardware, and the accessibility of equipment needed to manage the control station.

The design process of the control station began by obtaining dimensions of the electrical components. These dimensions were used to size the necessary compartments within the station. Figure 5.8 illustrates the final dimensions for the control station layout considering the space available in the back of the van.

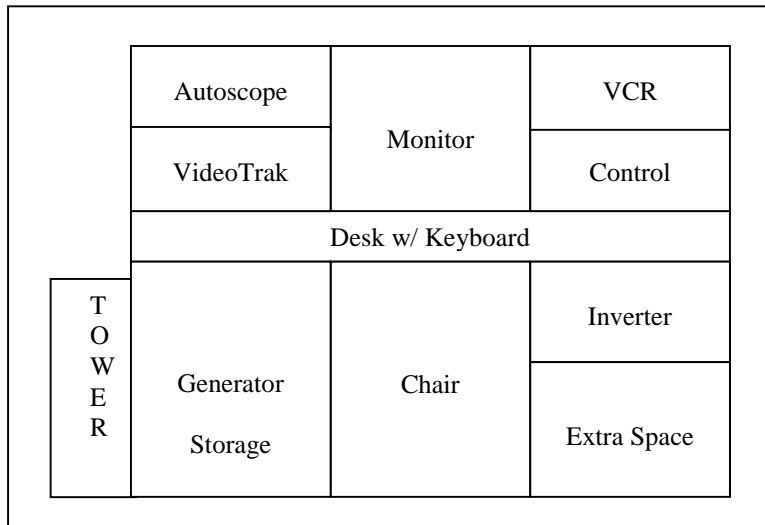


Figure 5.7 Control Station Layout (Facing Right Side of Van)

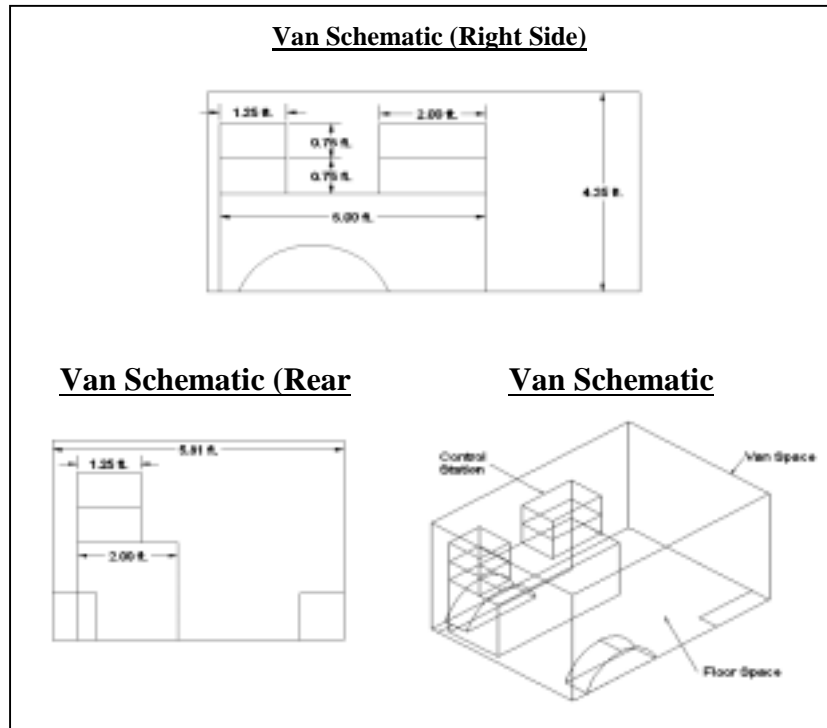


Figure 5.8 Van Schematic Layout

The racks were built with a combination of 90° angle and flat ¾” slot metal. The advantage of using slot metal is its high structural capacity, and its ability to be constructed in various ways. Using high-strength lock nuts and bolts the racking system was assembled to the specified dimensions obtained from the layout design. Since there were many pieces of slot metal used with different manufactured styles, the final framework of the rack did not exactly match dimensions from its layout, but was sufficient for the control station.

The slotted metal provided for a strong frame of the station, but proper shelving material would need to correspond with its unique attributes. The racks’ metal pieces were one inch in width. Accordingly, it was necessary the shelving provide for this empty space. Particleboard countertop was first considered since it possesses the properties of a smooth surface to work on and is lightweight. However, it was determined that the countertop would not be sufficient enough for the rack system. Therefore, it was decided

that one-inch sections of high-grade lumber would be used as the shelving. They were fastened to the rack with wood screws inserted in the slots of the metal rack. Admirable characteristics of the lumber include its ease of construction and the rough surface it provides for the components of the station.

The rack for the control station was nearly complete. There still needed to be a method devised for attaching the components secure to the rack. This would allow the components to occupy on the rack safely during the high acceleration and deceleration periods of the van's movement. Therefore, a strapping technique was employed, which is able to secure the individual components to their shelf. Lock straps are common tools found in hardware and home improvement stores. The straps are made of nylon and have a gear-lock system that provides essential restraint. Advertised functions of such straps include tying down equipment for hauling. The straps were integrated to the rack by first designating an area of shelving for each component, and then drilling ¼" inch slots about 2" in width where the straps could run through. After all necessary holes were drilled the racking system was ready to be installed into the van. Figure 5.9 illustrates the racking system for the control station that was placed inside the van. Its structural design offers accessibility, strength and sturdiness, and is able to manage the high acceleration-deceleration speed of the van, while providing safety.

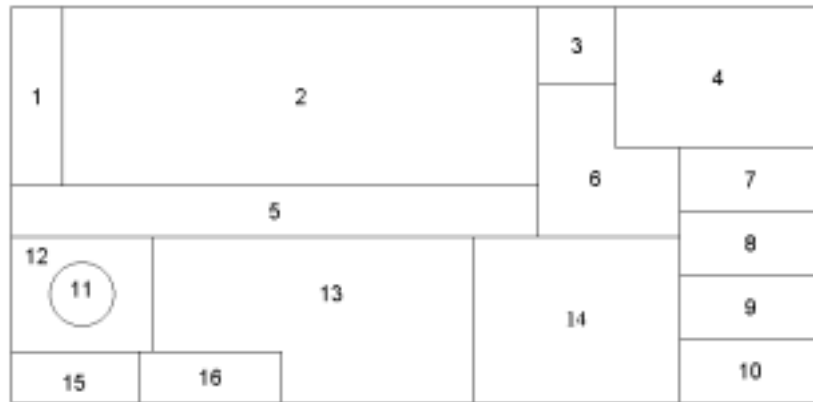
Preparation of the van for installation of the control station rack included the removal of its back seat. Consequently there were only two seats available for users, the driver and passenger seats. Also, because the van was a previous JTRP vehicle it had many scattered wires operating its safety light system. Some wires were removed while others were neatly arranged and placed to the side.



Figure 5.9 Portable Video Detection Control Station

The control station rack ultimately weighed 100 plus pounds and had a height comparable to the rear entrance of the van, which made it slightly difficult to place inside the van. Though its combination and size ultimately contributed to its safety within the van. First of all, the combination of height and width of the rack allows for only a small displacement, not full overturn if the van was involved in a forceful accident. In cases of high acceleration and deceleration periods of the van, it was thought that the rack could slightly move backwards or forward. This result was alleviated by affixing thick pieces of iron metal to the legs of the rack with $\frac{3}{4}$ " holes for reinforcement. The rack could then be fastened to the floor of the van with high-strength $\frac{3}{4}$ " nuts and bolts. Although this action was not accomplished, because a final inspection of the van flooring determined it was not structurally sound with its massive deposits of rust, and the gas tank of the van embodied most of the underside where the rack was situated. Ultimately the van with its control station rack not fastened to the floor was taken out for experimental trials, and it was found that the rack did not shift in high acceleration/deceleration situations.

The control station provides storage area for the major components of video detection while permitting an adequate area for the other supplies needed. These were specified in the complete list of components above. Figure 5.10 illustrates how those components and supplies are organized inside the van. It also represents how the complete portable video detection system is prepared for data collection in the field.



- | | | |
|------------------------|--------------------|------------------|
| 1. Computer Tower | 6. Open Space | 11. Safety Cones |
| 2. Control Station | 7. Concrete Block | 12. Wood Pieces |
| 3. Hydraulic Jack Unit | 8. Concrete Block | 13. Open Space |
| 4. Box – Control Wire | 9. Concrete Block | 14. Camera Mount |
| 5. Wooden Ladder | 10. Concrete Block | 15. Open Space |
| | | 16. Wheel Well |

Figure 5.10 Floor Layout of Portable Video Detection System

6. DATA COLLECTION

In the previous chapter a description of the mobile traffic laboratory was given to explain the technique of collecting videotape data. However, there was a need for a data collection plan before the mobile traffic laboratory could be used. This chapter discusses the data collection plan factors considered in this investigation, gives explanation on how the study intersections were selected, and describes how ground truth data was extracted.

6.1 Potential Factors of Video System Performance

Previous research has shown that there are numerous uncontrollable conditions that may affect the accuracy of vehicle detection. The uncontrollable factors that influence video detection include the presence of heavy vehicles, pedestrian traffic, video anomalies, and weather and light conditions. These intermediate factors can significantly influence the overall accuracy of the video detection systems. The factor that can be controlled is the camera position determined by camera offset and camera height. Proper camera position is critical to the successful performance of video detection. A properly placed camera accurately detects vehicles, maximizes the video detection systems capabilities, and determines the correct field-of-view. The factors investigated in this

study include: camera height, weather conditions, light conditions, intersection type, camera motion, presence of pedestrians, and the presence of heavy vehicles.

Camera Offset

Camera offset is defined as the horizontal distance from the camera to the center of the field-of-view in the image (center of intersection). It affects the performance of video detection through the phenomena called occlusion. Occlusion occurs when a vehicle or vehicles is hidden within a field-of-view by another object, usually a larger vehicle such as a semi-tractor trailer. Figure 6.1 demonstrates that a larger offset leads to greater probability of occlusion. In this study the effect of occlusion correlated with the size of the intersection. In all cases, the mobile traffic laboratory was parked at the corner of the intersections, and as close to the road as possible in concurrence with Indiana safety regulations. As a result, there is an indirect measurement of the occlusion effect through the size of the intersection.

Camera Height

The other factor that determines the level of occlusion is the camera height. A larger camera height decreases the probability of occlusion. The telescoping mast of Greenfield district had a maximum height of 25', while the mechanical mast on the

mobile traffic laboratory had a maximum height of 45'. Greenfield District acknowledged through its own study that a height of 25' is not sufficient to obtain an adequate field-of-view without occlusion. Therefore, two heights were chosen to collect the videotape data, 35' and 45'.

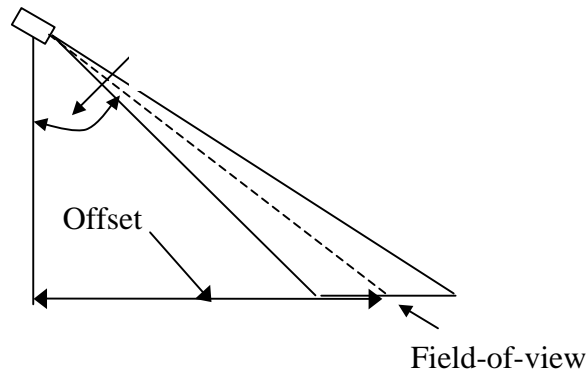


Figure 6.1 Camera Offset

Weather Conditions

It has been confirmed from past research that weather can affect the accuracy of video detection. Therefore, we tried to include dry, overcast, fog, rain, and snow weather conditions. However, during the time of data collection only dry, overcast, rain, and snow weather conditions were obtained. It should be noted that the mobile traffic laboratory was not utilized to obtain rain and snow conditions. Instead a mounted pan/tilt/zoom camera at a study intersection on Purdue campus was used.

Light Conditions

Light conditions can also affect video detection. Since, data collection may last up to sixteen hours, the counting period may include early morning hours, late evening hours, and night hours. During such a long period shadows and glare may occur. Previous research has shown that these light conditions affect the accuracy of video detection. Therefore, data was collected for three separate time periods: midday, evening, and nighttime. Midday conditions usually minimize the effects of long shadows and sun glare. Whereas, during evening conditions there is a day/night transition that creates the long shadow phenomena, and during night conditions vehicle headlights decreases the effectiveness of video detection.

Intersection Type

A portable video detection systems' function is to collect traffic data at any intersection. Hence, our study covered prevalent types of intersections. This study includes three common types of intersections: all-way stop control (AWSC), two-way stop control (TWSC), and signalized control (SC). Since the size of the intersection can affect video detection, larger intersections were covered recorded with two cameras.

Camera Motion

Camera motion can also affect the effectiveness of video detection. Camera motion is defined as light, medium, or heavy and is based on researchers observation.

Pedestrians

The presence of pedestrians within a field-of-view can also affect the effectiveness of video detection since they can be mistaken for vehicles. The number of pedestrians was documented for each approach of an intersection and later used in the analysis.

Heavy Vehicles

The objective of the portable video detection system is not only to count turning volumes at intersections, but also to classify vehicles. In addition, large heavy vehicles worsen the occlusion phenomenon. Therefore, the number of heavy vehicles was documented for each movement at the intersection, and then later used in the final analysis.

6.2 Data Collection Plan

Planning data collection began with the identification of potential intersections that the mobile traffic laboratory could collect videotape data. As previously stated, three types of intersections were investigated: the all-way stop-controlled, two-way stop-controlled, and signalized control. A comprehensive search of intersections in the Lafayette area, Indiana, brought a total of fifty intersections with a sufficient space to safely park the mobile traffic laboratory. Using the worksheet in Figure 6.2, parameters of each intersection were documented. These included preliminary observations of the level of traffic intensity, pedestrian intensity, and heavy vehicle intensity crudely evaluated as high, medium, or low. Also, the geometry of intersections was documented including the number of lanes, channelization, and median type, and then classified as large, medium, or small. Finally, six intersections were chosen for this study that represented diversity of all the intersections. Table 5.1 below describes the intersections with their corresponding factors along with illustrations shown in Figures 6.2 (a)-(h).

Intersection Type:	
Intersection Name:	
Space Available: (None, Some, Alot)	
Intersection Size: (Lanes, Median, Channelization)	
Traffic Intensity: (Low, Med, High)	
% Heavy Vehicles: (Low, Med, High)	
Pedestrians: (Low, Med, High)	

Figure 6.2 Data Inventory Worksheet

Table 6.1 Selected intersections

Intersection	Control	Size (# of Lanes)	Volume	Heavy Vehicles	Pedestrian Volume
Yeager/Cumberland	All-way	Medium (7)	Medium	Low	Low
US 231/600S	All-way	Small (4)	Medium	Med	Low
Yost Rd./S.R. 38	Two-way	Large (9)	High	Medium	None
CR 800/U.S. 52	Two-way	Large (8)	Medium	Medium	None
Northwestern/Stadium	Signalized	Large (9)	High	Medium	High
Kossuth/ Main St.	Signalized	Medium (6)	High	Medium	Low



Figure 6.3 Intersection Illustrations



Figure 6.3 Continued

6.3 Field Data Collection

As previously stated, the mobile traffic laboratory was used to record the videotape data. All data was videotaped during the summer and winter months of 2000. Except for one case, the mobile lab was setup at the corner of the intersections, usually at a 45° angle to both of the approaches. For one signalized intersection, videotape data was recorded utilizing a pre-existing mounted pan/tilt/zoom camera. This was done to collect videotape data during conditions of rain, snow, and nighttime. In all the cases, the mobile traffic laboratory and cameras were positioned to obtain the best field-of-view of the intersection. This included parking the mobile lab as close to the intersection as possible, and making certain that every turning movement was within the field-of-view while minimizing occlusion.

The videotape data was collected on two-hour Hi-8mm videotapes. These tapes could overlay the time and date on the image, which was later used to correlate ground truth counts with counts obtained from the video detection systems. During data collection for extended periods of time, the mast had to be lowered to exchange the videotapes, and then reset to its approximate position. While collecting data in the field, intersection, weather, and other environmental factors were documented using worksheets shown in Figures 6.4 and 6.5.

There were two problems that occurred during the data collection. The video camera used in this study had an auto-focus lens, which could not be remotely controlled. Subsequently, at two occasions of videotaping the camera became out of focus. This was during an exceedingly clear hot day and during nighttime. Explanation of the camera focus failing during the nighttime can be explained by the overwhelming light changes brought about by the headlights of vehicles. Therefore, nighttime data had to be collected using the pan/tilt/zoom camera. We could not find any convincing explanation of this phenomenon during the clear hot day.

INTERSECTION DATA COLLECTION WORKSHEET

Intersection Name (location) : _____

Height of Video Camera : 35 ft 45 ft

Intersection Type : _____

Initial Weather Conditions : Clear Partly Sunny Overcast Rain

Initial Wind Conditions : _____

Initial Light Conditions : High Noon Sunset Night

Number of Cameras Used: 1 2

Sketch of Observation Area (including dimensions)

Intersection Tape Log		
Tape #	Time	
	Begin	End

Comments

Figure 6.4 Intersection Data Collection Worksheet

INTERSECTION CONDITION OBSERVATION DATA SHEET

Intersection Name (location) : _____
 Intersection Type : _____
 Date of Observation : _____

Video Tape #	Time of Day	Weather Conditions	Light Conditions	Comments

Figure 6.5 Intersection Condition Observation Data Sheet

6.4 Ground Truth

Following the data collection effort, a comprehensive data inventory was completed. This included examining the data worksheets and tapes to develop the final format of data to be used in the analysis. Data extraction was separated into three stages. The first involved collecting interval turning movement counts using a Jamar manual counter. One-minute

counts were collected for movements of every approach at the intersection. This was accomplished by synchronizing the time of the Jamar unit with the time shown on the video images. Other data such as camera motion, weather conditions, and light conditions were also extracted for one-minute intervals. These data were then aggregated to the selected fifteen-minute interval of analysis.

Stage two of data extraction involved replaying the tapes to comment on phenomena that could affect video detection. These phenomena include such conditions as camera motion and the disruption in the quality of the videotape. In addition, the amount of pedestrians that could affect detector counts was documented. These include pedestrians that crossed any part of the pavement of an intersection, including the inside of the intersection. Furthermore, there was classification of vehicles for each movement into three categories: auto, single-unit truck, or semi-truck. An auto was considered as any car or truck that was not a single-unit truck or semi-truck. A single-unit truck was considered one that was perceived to have high payload. These include garbage trucks, autos pulling trailers, and truck with trailers that were not capable of separation. A semi-truck was considered as any tractor-trailer configuration where it was possible for the cab to be disconnected from the trailer itself. Once more, all of these details were separated per minute intervals to agree with the previous count intervals.

Finally, stage three involved arranging all recorded data into a standard format (Figure 6.6). This format was quintessential for efficiently managing the analysis portion of this research. The format first categorized data by movement (left, through, or right), and then by approach (north, south, east, west), and then by interval (1 to 7). Each data

point represented a fifteen-minute interval of count data. In addition, all other characteristics included in the analysis were given for each data point.

Interval	Type	Time	Height	PTZ	Cam	Dir	Light	Wind	Pric	Control	Lane N-S	Lane E-W	Lane Total	Approach	Movement	Ground Auto	Ground Single	Ground Truck	Ground Ped	Ground Vehicle	Percent Truck
1	100	2:53-3:07	45	0	1	NE	0	0	0	0	2	5	7	NB	LT	14	0	0	3	14	0
1	100	2:53-3:07	45	0	1	NE	0	0	0	0	2	5	7	NB	TH	13	0	1	3	14	7
1	100	2:53-3:07	45	0	1	NE	0	0	0	0	2	5	7	NB	RT	11	1	0	3	12	8
1	100	2:53-3:07	45	0	1	NE	0	0	0	0	2	5	7	SB	LT	5	0	0	0	5	0
1	100	2:53-3:07	45	0	1	NE	0	0	0	0	2	5	7	SB	TH	14	1	0	0	15	7
1	100	2:53-3:07	45	0	1	NE	0	0	0	0	2	5	7	SB	RT	4	0	0	0	4	0
1	100	2:53-3:07	45	0	1	NE	0	0	0	0	2	5	7	EB	LT	9	0	0	4	9	0
1	100	2:53-3:07	45	0	1	NE	0	0	0	0	2	5	7	EB	TH	23	0	0	4	23	0
1	100	2:53-3:07	45	0	1	NE	0	0	0	0	2	5	7	EB	RT	3	1	0	4	4	25
1	100	2:53-3:07	45	0	1	NE	0	0	0	0	2	5	7	WB	LT	6	1	0	0	7	14
1	100	2:53-3:07	45	0	1	NE	0	0	0	0	2	5	7	WB	TH	21	0	0	0	21	0
1	100	2:53-3:07	45	0	1	NE	0	0	0	0	2	5	7	WB	RT	5	0	0	0	5	0
2	100	3:08-3:22	45	0	1	NE	0	2	0	0	2	5	7	NB	LT	17	1	0	0	18	6
2	100	3:08-3:22	45	0	1	NE	0	2	0	0	2	5	7	NB	TH	22	2	0	0	24	8
2	100	3:08-3:22	45	0	1	NE	0	2	0	0	2	5	7	NB	RT	18	0	0	0	18	0
2	100	3:08-3:22	45	0	1	NE	0	2	0	0	2	5	7	SB	LT	4	0	0	0	4	0
2	100	3:08-3:22	45	0	1	NE	0	2	0	0	2	5	7	SB	TH	17	0	0	0	17	0
2	100	3:08-3:22	45	0	1	NE	0	2	0	0	2	5	7	SB	RT	1	0	0	0	1	0
2	100	3:08-3:22	45	0	1	NE	0	2	0	0	2	5	7	EB	LT	6	0	0	0	6	0
2	100	3:08-3:22	45	0	1	NE	0	2	0	0	2	5	7	EB	TH	25	0	0	0	25	0
2	100	3:08-3:22	45	0	1	NE	0	2	0	0	2	5	7	EB	RT	12	0	0	0	12	0
2	100	3:08-3:22	45	0	1	NE	0	2	0	0	2	5	7	WB	LT	7	0	0	3	7	0
2	100	3:08-3:22	45	0	1	NE	0	2	0	0	2	5	7	WB	TH	32	1	0	3	33	3

Figure 6.6 Example Data Format for Analysis

7. AUTOSCOPE METHOD AND EVALUATION

The Autoscope video detection system is explained in Chapter 5. Autoscope is considered a trip-wire detection system. Trip-wire systems detect vehicles selected areas within a video image, while the rest of the image is ignored. These areas are called “detection zones”. Vehicles are detected when they pass through a detection zone by pixel intensity change within the detection zone. In other words, Autoscope provides spot detection of vehicles within a video image.

This chapter explains the method and evaluation procedures used to analyze data with the Autoscope video detection system. During the research two concepts of estimating turning volumes were considered. The first was a tracking concept that used Autoscope detectors arranged in a matrix pattern. In the course of research, this method was abandoned because of the problems the Autoscope system had in handling the matrix concept. Subsequently, a different method was devised that was based on flow conservation. According to the flow conservation principle, what enters an intersection must exit, to conserve traffic flow within the intersection. Initial simulation tests of this method gave promising results and researchers proceeded with this concept.

This chapter first introduces the detector matrix method, and gives an explanation why it was abandoned in this research. Subsequently, there is a detailed explanation of the traffic conservation method. It describes the formulation of the method, and simulation analysis used to improve it. Next, the extraction of data with Autoscope is

described. Finally, the statistics used to analyze the Autoscope data are introduced, the results are reported and conclusions are made from the results.

7.1 Detector Matrix Method

Our first idea was to use Autoscope detectors to track vehicles within an intersection. A rational approach was to place detectors along the trajectory of vehicles to track their individual movements. However, this was time-consuming since a new detector layout would have to be created for every intersection. Therefore, it was decided to use a matrix of detectors that could be able to track movements at any intersection. Autoscope output from the matrix of detectors in conjunction with an additional post processing method would be used to estimate turning movements.

The concept of using a matrix of detectors allows for any vehicle entering the field of view of the camera to be tracked throughout the matrix. The flexibility of the matrix allows for any intersection to be analyzed. Hence, the versatile matrix can be used for any video image. Detectors not within the intersection should be removed because they can cause false detections that can affect Autoscope's ability to report detections. In addition, the relative position and size of the detectors can be adjusted to fit any particular geometry of an intersection. This is important because there are an unlimited number field-of-views for an intersection. The proposed matrix (Figure 7.1) contains the maximum number of ninety-nine detectors that the Autoscope software allows for a detector layout.

This method uses Autoscope's ability to output individual 'event' data, synonymous to per vehicle record data obtained from inductive loops. The relevant information needed from event data output includes vehicle arrival times, occupancy times, and corresponding detectors' IDs. The vehicle arrival time is the time when a vehicle enters the detection zone. The format of the vehicle arrival time is in hour, minute, and seconds, where seconds are reported in the thousandths. The occupancy time is the total time a vehicle occupies a detector, which is also reported in thousandths of seconds. The detector ID is a random number assigned to each detector upon placement within a detector layout and can be used for identification.

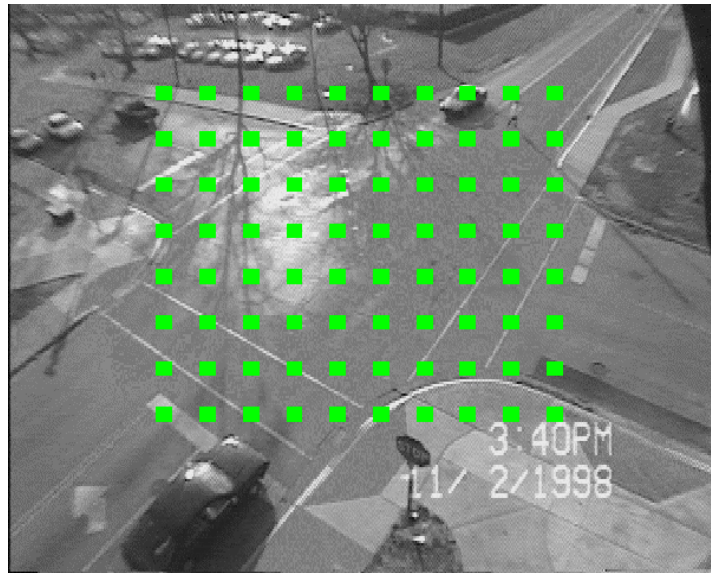


Figure 7.1 Example Detector Matrix Layout

A technique using these recorded events was developed to track vehicles. The basic idea was to identify the events made by an individual vehicle through the matrix of

detectors. Therefore, criteria for tracking events within the matrix were developed to obtain necessary turning movements. There are two requirements for a detectors' configuration within a matrix to track vehicles:

1. The distance between two consecutive detectors must be shorter than the length of the shortest vehicle;
2. The detector width must be shorter than the shortest distance between consecutive vehicles.

The first requirement makes it theoretically possible to track individual vehicles throughout the matrix as the vehicle should be activating at least two detectors at a time. As stated earlier, the activation of the detector constitutes an event. Hence, the events from consecutive detectors for the same vehicle can be correlated. Events that are consecutive should possess detection occupancy times that overlap, and then can be associated to an individual vehicle. An illustration of the overlapping of detector occupancy times is shown in Figures 7.2. As a vehicle exits Detector 1, it must enter the Detector 2, since the distance between these two detectors is shorter than the vehicle length. Therefore, the Detector 2 event-time begins before the Detector 1 occupancy-time terminates, which results in two detectors on at the same time. Hence, an individual vehicle should be able to be tracked throughout the matrix.

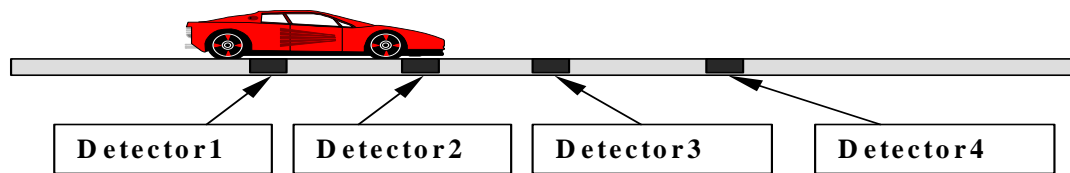


Figure 7.2 Example of Requirement 1

The second requirement ensures that all vehicles are detectable. Figure 7.3 illustrates an example where the width of Detector 2 exceeds the minimum gap between two consecutive vehicles. After the first vehicle enters the Detector 2 activation zone, the detector's occupancy time will not terminate before the second vehicle enters that same detector. The two vehicles simultaneously occupy the detector and a single occupancy time represents both the vehicles' path. In this case, the second vehicle's arrival time to Detector 2 is not recorded, and it would be assumed that only one vehicle, the first vehicle, was detected for a longer period of time. Hence, the detector width should be shorter than the shortest distance between consecutive vehicles, but not too small to maintain the detector's detection capability.

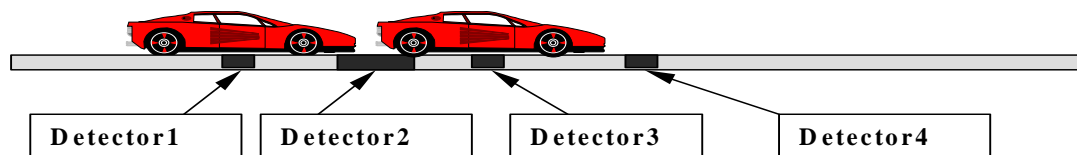


Figure 7.3 Example of Requirement 2

Autoscope event data in conjunction with an additional post processing method was needed to estimate turning movements. Therefore, based on the two previous requirements, the following criterion was used to develop an algorithm. Event-time(i) and Event-time(i+1) represent the same vehicle if:

1. $\text{Event-time}(i) \leq \text{Event-time}(i+1)$;
2. $\text{Event-time}(i) + \text{Occupancy-time}(i) \geq \text{Event-time}(i+1)$.

Event-time(i) is the vehicle arrival time at detector(i), occupancy-time(i) is the time when detector(i) is occupied by the vehicle, and event-time(i+1) is the vehicle arrival time at detector(i+1). Using these criteria to track vehicles within a matrix of Autoscope detectors, an algorithm was developed to post process the Autoscope event data to obtain turning movement counts at an intersection.

In the process of coding the algorithm, initial tests on the capabilities of the matrix were performed. These tests indicated a severe problem for Autoscope to process the massive amounts of data from an intersection video image. There were many instances of random, excluded, and repeated detections. This is attributed to the detector load index of the matrix on Autoscope. Essentially, the load index reflects the quantity of data Autoscope needs to process. The higher the load index, the greater for the Autoscope MVP to either miss or create false detections. Furthermore, the matrix performance during certain light conditions was poor. Consequently, it was decided to abandon this method and devise another concept.

7.2 Flow Conservation Method

Two approaches for estimating turning movements using spot detection are presented in Chapter 3. The first approach includes methods useful when the number of turning volumes is greater than the number of detectors. Typically, eight spot detectors for counting entrance and exit flows are used. Other sources of information and assumptions, such as turning percentages, are needed to estimate the turning volumes. Since video detection allows for placement of a large number of detectors, this approach can be further developed. The idea is to use the flow conservation assumption to estimate turning volumes from multiple flow counts and then utilize data redundancy to improve the estimation accuracy.

The method will be demonstrated with an example of a four-leg intersection with no turning bays (Figure 7.4). This type of intersection is probably the most difficult to obtain turning movements since vehicles are unable to use exclusive turning lanes. In addition to the placement of eight detectors at the entrance and exit of the approaches, eight other detectors are placed within the intersection. The detection spots should be selected as areas with little or no stopped vehicles. This increases the counting accuracy of the individual detectors.

Figure 7.4 illustrates that individual detectors are capable of counting multiple flows. However, if a flow is assigned to a detector, all vehicles of the flow should pass over the detector. This condition allows assigning entire flows to one or more detectors (e.g. Flow 2 is assigned to Detectors 1, 3, 4, and 7). Flow and detector notation is given in Table 7.1, and the detector-flow matrix assignment is shown in Figure 7.5.

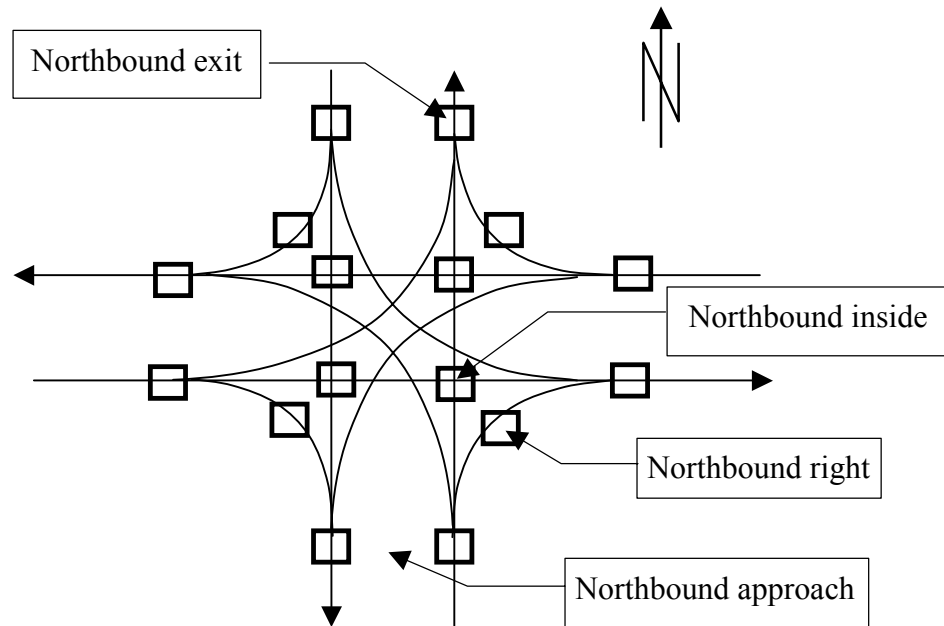


Figure 7.4 Example Intersection with detector locations

Table 7.1 Numbering of the detectors and flows at a four-leg intersection

Approach	Movement	Flow	Detector	Count
Northbound	Right turn	F_1	Approach	D_1
	Through	F_2	Right turn	D_2
	Left turn	F_3	Inside	D_3
			Exit	D_4
Westbound	Right turn	F_4	Approach	D_5
	Through	F_5	Right turn	D_6
	Left turn	F_6	Inside	D_7
			Exit	D_8
Southbound	Right turn	F_7	Approach	D_9
	Through	F_8	Right turn	D_{10}
	Left turn	F_9	Inside	D_{11}
			Exit	D_{12}
Eastbound	Right turn	F_{10}	Approach	D_{13}
	Through	F_{11}	Right turn	D_{14}
	Left turn	F_{12}	Inside	D_{15}
			Exit	D_{16}

		NL	NT	NR	WL	WT	WR	SL	ST	SR	EL	ET	ER
		F ₁	F ₂	F ₃	F ₄	F ₅	F ₆	F ₇	F ₈	F ₉	F ₁₀	F ₁₁	F ₁₂
N_APPR	D ₁	1	1	1	0	0	0	0	0	0	0	0	0
N_RIGHT	D ₂	0	0	1	0	0	0	0	0	0	0	0	0
N_INSIDE	D ₃	0	1	0	0	0	0	0	0	0	0	1	0
N_EXIT	D ₄	0	1	0	0	0	1	0	0	0	1	0	0
W_APPR	D ₅	0	0	0	1	1	1	0	0	0	0	0	0
W_NRIGHT	D ₆	0	0	0	0	0	1	0	0	0	0	0	0
W_INSIDE	D ₇	0	1	0	0	1	0	0	0	0	0	0	0
W_EXIT	D ₈	1	0	0	0	1	0	0	0	1	0	0	0
S_APPR	D ₉	0	0	0	0	0	0	1	1	1	0	0	0
S_RIGHT	D ₁₀	0	0	0	0	0	0	0	0	1	0	0	0
S_INSIDE	D ₁₁	0	0	0	0	1	0	0	1	0	0	0	0
S_EXIT	D ₁₂	0	0	0	1	0	0	0	1	0	0	0	1
E_APPR	D ₁₃	0	0	0	0	0	0	0	0	0	1	1	1
E_RIGHT	D ₁₄	0	0	0	0	0	0	0	0	0	0	0	1
E_INSIDE	D ₁₅	0	0	0	0	0	0	0	1	0	0	1	0

Figure 7.5 Detector-flow assignment matrix for the example intersection

The assumption of the flow conservation method is that vehicles defined for a flow must pass over all the detectors to which that flow is assigned. However, if data is collected from detectors during similar time periods, it is possible there is only an approximation of the flow conservation assumption. These include cases for vehicles that are inside the intersection during the beginning and end of the counting periods, and when counting periods are shorter than travel times between detectors. Therefore, the counting periods should be several minutes long. In most traffic studies, 15-minute counting intervals or longer are used.

7.2.1 Estimation Method

Individual detectors counting vehicles belonging to designated flows can be described as:

$$D_i = \sum_j a_{ij}F_j + \varepsilon_i \quad (7.1)$$

where:

D_i = detector count i , $i = 1 \dots n$;

F_j = turning flow j , $j = 1 \dots n$;

a_{ij} = detector-flow assignment matrix, $a_{ij}=1$ if detector i counts flow j ; = 0 otherwise;

ε_i = counting error for detector i .

Equation 7.1 can be solved using any type of regression. However, there was investigation into the error term before a proper regression technique was proposed. The error term ε has zero mean and non-zero variance and is considered as random error. There are two possible sources of count errors that can occur; missed detections and false detections. In the first case, a vehicle is not detected and the corresponding error is -1. In the second case, the detector returns a multiple count when one vehicle passes over the detection zone. The detection error is then the multiple count minus one. Now let us assume that the likelihood of the first error is p , the likelihood of the second error is q , and the average multiple count is n . Then, the expected error associated with one vehicle passage would be $(1-p-q) \cdot 0 + p \cdot (-1) + q \cdot n = q \cdot n - p$, and since the independent variances sum up, the corresponding total variance is approximately $(1-p-q) \cdot 0^2 + p \cdot (-1)^2 + q \cdot n^2 = (p + q \cdot n^2)$.

There are three assumptions in the detector-count error term. First, we assumed that the variance of the error is approximately the same for each vehicle. In addition, we assumed that the errors occur independently one from another. Finally, the third assumption was that counting errors not associated with vehicle passage are negligible (includes vehicles of other flows, pedestrians, other objects, etc.). These assumptions allow expressing the variance of counting error as the sum of variances generated by D individual vehicles:

$$\text{var } \varepsilon = D(p + qn^2) \quad (7.2)$$

Equation 7.2 indicated that the standard error grew proportionally to \sqrt{D} . In order to use a simple regression to solve for Equation 1, the model was transformed by dividing both its sides by \sqrt{D} . The resulted equation is:

$$\sqrt{D_i} = \sum_j \frac{a_{ij}}{\sqrt{D_i}} F_j + \frac{\varepsilon_i}{\sqrt{D_i}}. \quad (7.3)$$

Unlike the ε error in Equation 7.1, the ε/\sqrt{D} error in Equation 7.3 was believed to have uniform variance across detectors. The ordinary least-square regression was now suitable to find turning flows F_j using Equation 7.3.

7.2.2 Simulation Test

A simulation test was done to reveal issues that were not anticipated during the development of the method's concept. Detector counts were first simulated for assumed turning flows given in the third column of Table 7.1 with the example detector layout shown in Figure 7.4. Then so-called ideal detector counts were calculated using the assumed turning flows and the detector-flow assignment matrix shown in Figure 7.5. Ideal detector counts are free of any count errors. Next, the ideal detector counts were contaminated with random errors to simulate actual detector counts that are imperfect. The error for a particular detector count D was assumed to be $D^{0.5} \cdot \epsilon$, where ϵ was the error randomly selected from the range between -3 and 3. These error limits were assumed arbitrarily. According to the properties of the uniform distribution, the variance of the error was $3 D$. This was consistent with the error structure formulated in Equation 7.2.

Detector counts contaminated with errors were then simulated one hundred times. Table 7.2 shows the ideal detector counts and summarizes the ranges and standard deviations of the corresponding simulated detector counts. Ordinary regression was then used with the model in Equation 7.3 to estimate the turning flows from each set of simulated detector counts. Table 7.2 summarizes the obtained turning flow estimates. As expected, the turning flow estimates seemed to be unbiased as well as the estimated standard errors.

The fourth column in Table 7.2 gives so-called reference standard errors of estimation. These were obtained for cases where each turning flow is measured with its

own detector. Similar to the detector counts, the variances of these errors were $3 \cdot F$. It can be seen in Table 7.2 that the proposed method was more efficient in estimating the through flows than the direct counting of these flows. The proposed method was only marginally better than the direct counting for right turns, and was much worse for left turns. The results reflected the information on the turning flows given by Equations 7.1 and 7.3. Four detectors counted each through flow. Three detectors, one of which exclusively counted right turns only, counted each right turning flow. Only two detectors together with two other flows counted each left turning flow. Therefore, it was concluded that the quality of estimates depended on the number of independent detectors used and the number of flows assigned to those detectors.

Table 7.2 Effect of incorrect weights on the regression results ($D^{0.5}$ is correct)

Mvnt	Flow	Ideal Count	Ref. Std. Err	Weight = $D^{0.5}$			Weight = 1			Weight = D		
				Mean Est.	Act. Std. Err.	Reg. Std. Err.	Mean Est.	Act. Std. Err.	Reg. Std. Err.	Mean Est.	Act. Std. Err.	Reg. Std. Err.
NL	F ₁	200	24	201	54	54	203	50	25	195	48	25
NT	F ₂	500	39	498	31	36	494	33	7	501	33	6
NR	F ₃	300	30	302	28	29	299	29	10	302	31	10
WL	F ₄	180	23	179	45	53	180	46	26	186	41	22
WT	F ₅	450	37	445	28	37	450	36	8	447	31	7
WR	F ₆	270	28	268	27	29	273	26	10	270	27	10
SL	F ₇	160	22	165	40	51	156	42	27	152	44	29
ST	F ₈	400	35	400	29	35	403	29	7	398	29	7
SR	F ₉	240	27	242	24	26	239	31	13	243	24	10
EL	F ₁₀	140	20	143	39	51	143	43	8	145	46	25
ET	F ₁₁	350	32	347	27	35	352	30	22	350	28	18
ER	F ₁₂	210	25	212	26	28	209	25	11	208	25	8

There were strong assumptions made about the structure of the error term in Equation 7.2. In fact, the actual error variance could be different from the assumed in the model. Therefore, the effect of incorrect variances was tested using three different scaling weights: 1, $D^{0.5}$, and D . For all three cases, the correct weight was $D^{0.5}$. The results obtained for all the three cases are shown in Table 7.2, and the following expectations were confirmed:

1. Use of incorrect weights does not introduce any bias to the flow estimates.
2. Effectiveness of estimation is comparable in all three cases.
3. The standard errors are considerably underestimated in the two cases of incorrect weights.

7.2.3 Preliminary Regression Analysis

The purpose of this analysis was to investigate the assumptions made of the Flow Conservation method as presented in Chapter 7.2, and to verify results obtained from simulation in Chapter 7.2.2. In addition, the ability of selecting a sufficient number of spots traversed by certain turning flows had to be checked for a real intersection using video detection methods. It involved analyzing three videotapes with data extracted in thirty-minute intervals using procedures as described in Chapter 7.3.

Figures 7.6 and 7.7 show the preliminary results from the analysis. It demonstrated that the proposed Flow Conservation method was feasible for video

detection, and it was possible to meet the detector-flow assignment requirements real intersections on a video image. In addition, Figure 7.6 proved that there was no homoscedasticity of the detector error as initially thought in Equation 7.2 of Chapter 7.2.1. Alternatively, the error remained fairly uniform, or homogenous. Therefore, simple linear regression could be used without the \sqrt{D} transformation as proposed in Equation 7.2. Therefore, it was decided to utilize Equation 7.1 as the simple linear regression equation for future analysis

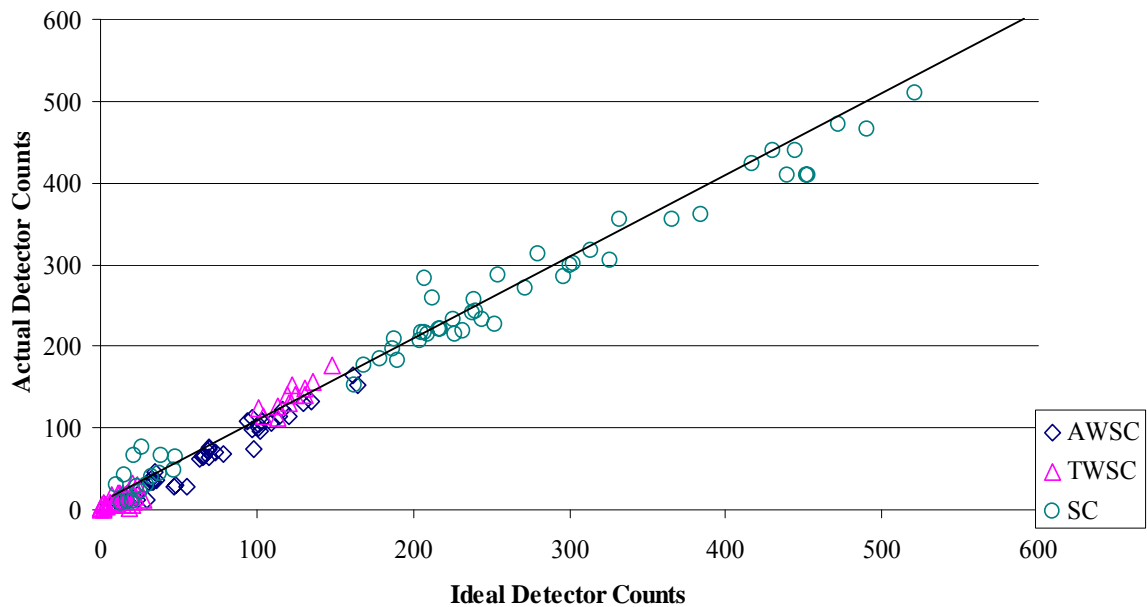


Figure 7.6 Actual detector counts vs. ideal detector counts

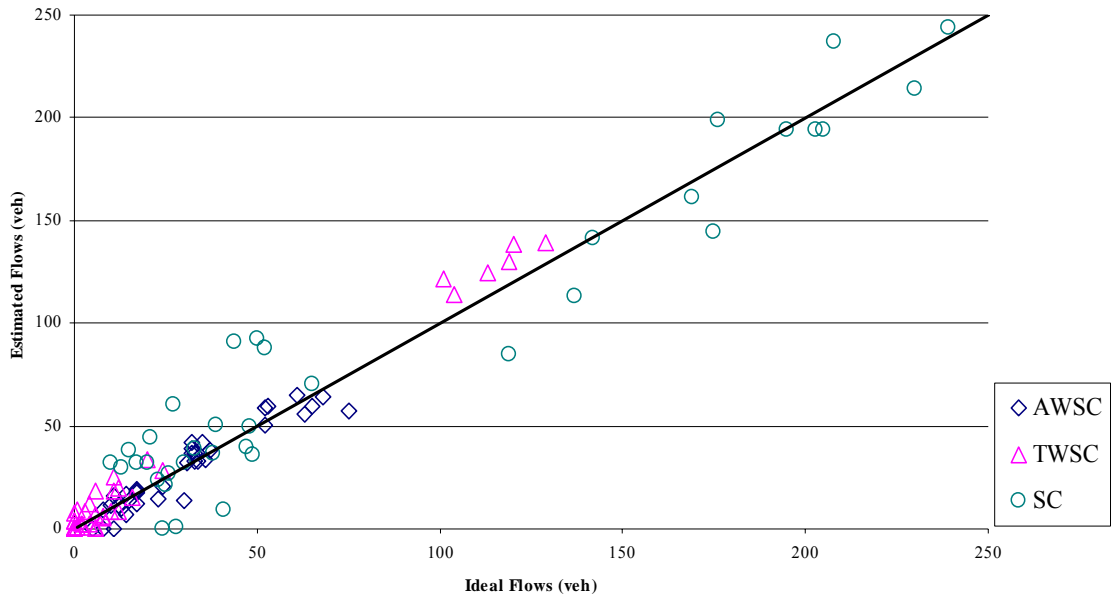


Figure 7.7 Turning Flow estimates vs. ground truth flows

7.3 Data Extraction

The Autoscope 2004 was used to retrieve 15-minute data from the videotapes automatically. This entailed creating detector layouts for all the video images obtained during field data collection. The detector layouts differed across intersections by number, type, and size of the detection zones used. We attempted to set detector layouts such that all vehicles of a certain turning flow traversed a particular detection spot as stipulated from our method. To accomplish this objective, directional presence and count detectors were used, along with the logical AND function for multiple detectors. In addition, detector stabilizers were used to reduce detector errors caused by camera motion.

Detector stabilizers are able to sense movement of a video image, and then compensate that movement into the detectors. Examples of such detector layouts used for analysis are shown in Figure 7.8.

The process of obtaining Autoscope data first began by developing the detector layouts. The detector layouts consisted of detectors, detectors stabilizers, and detector stations. The detector stations are used when collecting interval data with Autoscope. The detector stations were set to accumulate data in one-minute intervals to correspond with the ground truth data. In addition, the Autoscope date and time were set to match with the display given on the video images. Subsequently, the detector layouts were then analyzed using the Autoscope unit. During Autoscope data extraction, several detectors were noticed of giving frequent multiple and false detections. These were either adjusted to find a better location and size, or deleted because despite our effort to find them the best location and size they would still give false detections.

The output of the interval data came in the form of text files. These text files were then imported into a spreadsheet application for manipulation where selected fifteen-minute interval counts could be aggregated for all the data. These intervals matched with aggregated fifteen-minute counts from the ground truth data. Subsequently, detector-matrices were developed for the individual detector layouts. Then ordinary regression was used to estimate flows as presented in the traffic conservation method. Finally, the estimated flows were appended to the standard format of data used in the analysis (e.g. Table 7.3), along with the count error and square count error, which was used later in analysis.



(b) Yeager/Cumberland (AWSC)



(c) YostRd./SR 38 (TWSC)



(h) Northwestern/Stadium (SC)

Figure 7.8 Examples of detector layouts

Table 7.3 Example of Extracted Autoscope Data

Interval	Tape	Time	•	Autoscope Vehicle	Autoscope Error	Autoscope Square Error
1	100	2:53-3:07	•	21	7	49
1	100	2:53-3:07	•	9	-5	25
1	100	2:53-3:07	•	22	10	100
1	100	2:53-3:07	•	4	-1	1
1	100	2:53-3:07	•	11	-4	16
1	100	2:53-3:07	•	0	-4	16
1	100	2:53-3:07	•	8	-1	1
1	100	2:53-3:07	•	27	4	16
1	100	2:53-3:07	•	6	2	4
1	100	2:53-3:07	•	10	3	9
1	100	2:53-3:07	•	36	15	225
1	100	2:53-3:07	•	0	-5	25

7.4 Method Evaluation

This research is concerned with the evaluation of the performance of video detection to count turning movements at an intersection. Therefore, the counting error was used to evaluate both video detection systems. The counting error in Equation 7.4 was considered the difference between the video detection count and the ground truth count. It is able to describe both the magnitude and direction of the error of each video detection system. Descriptive statistics were developed to describe this counting error. These equations include: mean true count (7.5), mean error (7.6), standard error (7.7)

relative mean error (7.8), and relative standard error (7.9). Below are equations that formulate the descriptive statistics used in this study:

$$\text{Counting Error (CE}_i\text{)} = \text{VC}_i - \text{GC}_i; \quad (7.4)$$

$$\text{Mean True Count (MC)} = \frac{\sum \text{GC}_i}{N}; \quad (7.5)$$

$$\text{Mean Error (ME)} = \frac{\sum \text{CE}_i}{N}; \quad (7.6)$$

$$\text{Standard Error (SE)} = \sqrt{\frac{\sum \text{CE}_i^2}{N}}; \quad (7.7)$$

$$\text{Relative Mean Error (RME)} = \frac{\text{ME}}{\text{MC}}; \quad (7.8)$$

$$\text{Relative Standard Error (RSE)} = \frac{\text{SE}}{\text{MC}}; \quad (7.9)$$

where: N = number of observations.

The above descriptive statistics were used for both video detection systems. They are able to quantify the counting error resulted from either system. In addition, the affect of environmental and weather on turning movement count estimation was evaluated. This

was accomplished using additive linear regression models. There were two additive regression models developed for the data collected with the mobile traffic laboratory, and the data collected with the PTZ camera. There needed to be two separate models because each set of data comprised of different descriptive variables to be analyzed. This would have not been the case if all videotape data were collected with the mobile traffic laboratory.

The first regression model in Equation 7.10 was used for the videotape data collected from the mobile traffic laboratory. The second regression model in Equation 7.11 was used for the videotape data collected from the pan/tilt/zoom camera. Both models have similar factors that were analyzed, but the most important difference is the second model describes the weather effects of rain, snow, and nighttime. Most of the explanatory variables were binary variables while the others were continuous variables. The statistical modeling was performed using the Statistical Analysis Software (SAS, 2000)

$$|CE| = \beta_0 + \beta_1 Hei + \beta_2 Cam + \beta_3 Lmid + \beta_4 Leve + \beta_5 Wmod + \beta_6 Wheav + \beta_7 Twsc + \beta_8 Sig + \beta_9 Le + \beta_{10} Ri + \beta_{11} La + \beta_{12} Auto + \beta_{13} Sing + \beta_{14} Truck + \beta_{15} Ped \quad (7.10)$$

$$|CE| = \beta_0 + \beta_1 Ni + \beta_2 Le + \beta_3 Ri + \beta_4 Rain + \beta_5 Snow + \beta_6 Auto + \beta_7 Sing + \beta_8 Truck + \beta_9 Ped \quad (7.11)$$

where:

- $|CE|$ = absolute value of the counting error of video detection system in veh/15-min;

- Hei = indicator variable for height of camera; Hei = 0 when height = 35' and Hei = 1 when height = 45';
- Cam = indicator variable for number of cameras; Cam = 0 when camera = 1 and Cam = 1 when camera = 2;
- Lmid = indicator for midday light conditions; Lmid = 0 when light was overcast and Lmid = 1 when light was midday;
- Leve = indicator for evening light conditions; Leve = 0 when light was overcast and Leve = 1 when light was evening;
- Wmod = indicator for moderate wind conditions; Wmod = 0 when wind was light and Wmod = 1 when wind was moderate;
- Wheav = indicator for heavy wind conditions; Wheav = 0 when wind was light and Wheav = 1 when wind was heavy;
- Twsc = indicator for two-way stop control; Twsc = 0 when all-way stop control and Twsc = 1 when two-way stop control;
- Sig = indicator for two-way stop control; Sig = 0 when all-way stop control and Sig = 1 when signalized control;
- Le = indicator for left turn movement; Le = 0 when through movement and Le = 1 when left movement;
- Ri = indicator for right turn movement; Ri = 0 when through movement and Ri = 1 when right movement;
- La = number of lanes of the intersection minus four;
- Auto = number of auto vehicles;
- Sing = number of single unit trucks;
- Truck = number of semi-trailer trucks;
- Ped = number of pedestrians;
- Ni = indicator for night conditions; Ni = 0 when light is overcast and Ni = 1 when light is night;

- Rain = indicator for rain conditions; Rain = 0 when light is overcast and no rain and Rain =1 when there is rain;
- Snow = indicator snow conditions; Snow = 0 when light is overcast and no snow and, TWSC =1 when there is snow.

7.4.1 Autoscope Descriptive Statistics Results

Results from the entire data set including the data collected with the Mobile Traffic Laboratory and with the mounted pan/tilt/zoom camera are shown in Table 7.4 and Figure 7.9. There were a total of 2303 observations with an average vehicle count of 25.7. There was an overestimated mean error of 4.0 with standard error of 16.8. However, relative to the average vehicle count, the Autoscope system overestimated 15.4% of the turning movements with standard error of 65.3%. Table 7.4 also breaks down the errors according to specific intersection. Among all the three intersections, the signalized one had the best turning movement estimations.

Table 7.5 shows results obtained from data collected only from the Mobile Traffic Laboratory. Overall, there were 1764 observations with an average vehicle count of 21.4 veh/15-min. There was a mean error of 3.0 veh/15-min. with standard error of 16.1 veh/15-min. Relative to the average vehicle count; the Autoscope system overestimated 14.2% of the turning movements with relative standard error of 75.4%. Table 7.5 also breaks down the errors according to specific interval, height, and number of cameras, camera direction, light, camera motion, traffic control and movement.

Table 7.6 and Figure 7.9 shows results obtained from data collected only from the mounted pan/tilt/zoom (PTZ) camera. Overall, there were 539 observations with an average vehicle count of 39.7 veh/15-min. There was an overestimated mean error of 7.0 veh/15-min. with standard error of 18.7 veh/15-min. Relative to the average vehicle count; the Autoscope system overestimated 17.6% of the turning movements with standard error of 47.1%. Table 7.6 again breaks down error according to specific interval, light conditions, precipitation and movement.

Table 7.4 Descriptive Statistics for Entire Data set (Mobile Laboratory and PTZ)

Location	No. of Lanes	Traffic Control	No. of Obs.	Average Count	Mean True Error	Standard Error	Relative Mean Error (%)	Relative Standard Error (%)
Old U.S. 231/ CR500S	4	AWSC	168	7.3	2.1	3.9	28.9	53.3
Cumberland/ Yeager Road	7	AWSC	576	13.0	1.3	14.8	9.7	114.2
U.S. 52/ CR 400S	8	TWSC	168	12.1	8.2	18.8	68.2	156.2
SR 38/ Yost Road	9	TWSC	348	15.8	3.5	11.8	21.9	74.5
Salisbury St. / Stadium Ave.	9	Signal	539	39.7	7.0	18.7	17.6	47.1
Main Street/ Kossuth	6	Signal	504	42.8	3.4	21.1	7.9	49.3
Total			2303	25.7	4.0	16.8	15.4%	65.3%

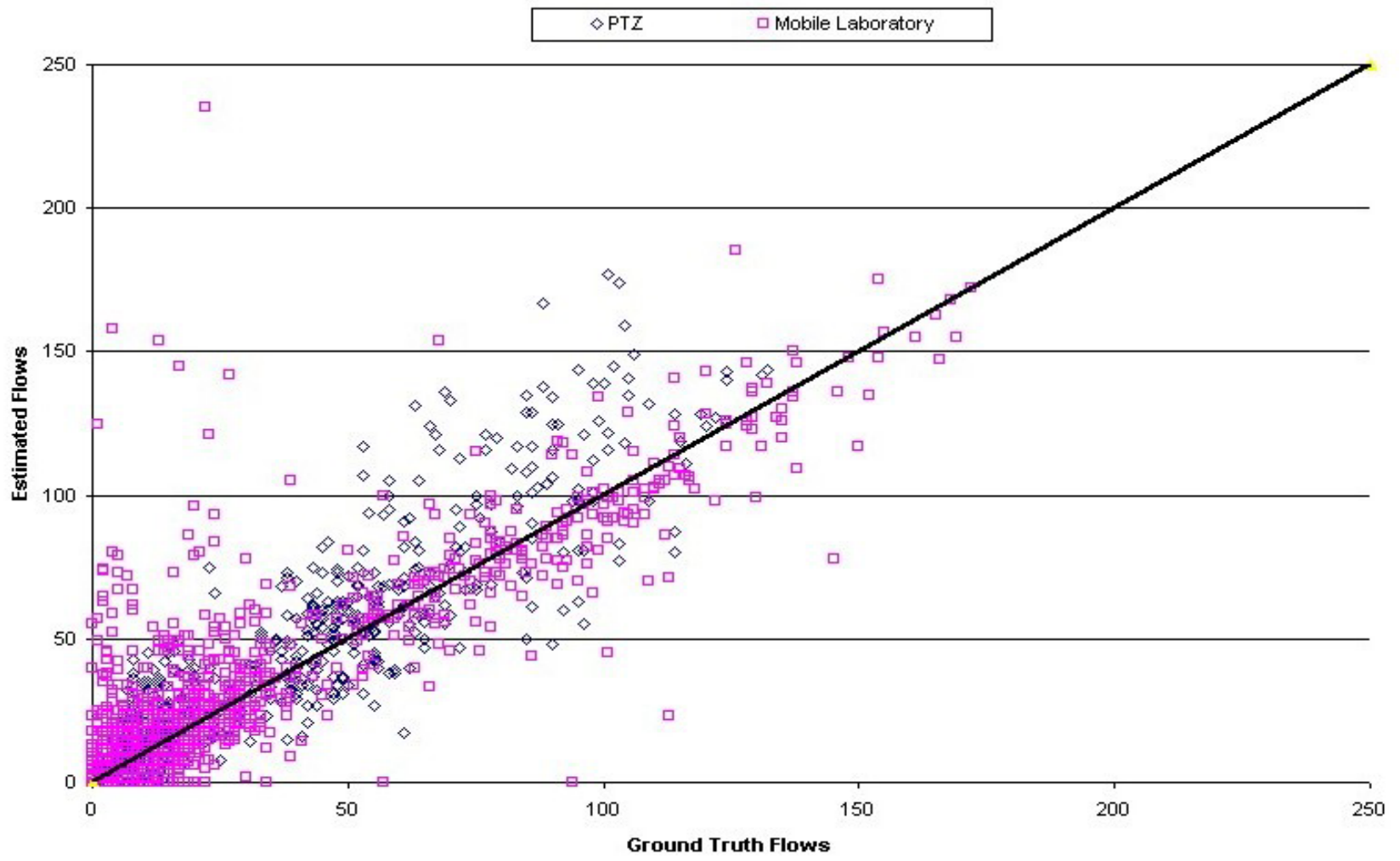


Figure 7.9 Estimated Flows vs. Ground Truth

Table 7.5 Descriptive Statistics for Data collected with Mobile Laboratory

Characteristic		Number of Observations	Average Count	Mean Error	Standard Error	Relative Mean Error (%)	Relative Standard Error (%)
Interval Number	1	264	21.0	2.9	16.9	13.6	80.5
	2	252	21.1	2.0	10.0	9.5	47.5
	3	252	21.1	2.3	11.3	10.7	53.5
	4	252	21.9	3.6	21.7	16.3	99.0
	5	252	21.0	2.8	15.2	13.3	72.1
	6	252	21.7	3.6	16.5	16.7	75.8
	7	240	22.0	4.2	18.5	19.2	84.1
Height (ft.)	35	336	30.6	3.0	23.0	9.9	75.1
	45	1428	19.2	3.0	14.1	15.8	73.1
No. of Cameras	1	1092	13.0	2.1	12.7	16.1	97.9
	2	672	35.1	4.6	20.5	13.1	58.5
Direction	E	252	42.4	3.4	21.2	8.0	49.9
	N	252	43.2	3.4	21.0	7.8	48.6
	NE	576	13.0	1.3	14.8	9.7	114.2
	NW	168	7.3	2.1	3.9	28.9	53.3
	SW	348	15.8	3.5	11.8	21.9	74.5
	W	168	12.1	8.2	18.8	68.2	156.2
Light	Midday Sun	1152	23.4	2.9	15.9	12.4	68.0
	Evening Sun	420	20.9	4.7	19.6	22.6	93.8
	Overcast	192	10.9	0.2	6.7	2.2	61.4
Camera Motion	Weak	1176	17.8	2.7	15.4	14.9	86.4
	Moderate	396	27.1	5.0	19.0	18.5	70.1
	Heavy	192	31.6	1.3	14.1	4.0	44.6
Traffic Control	AWSC	744	11.7	1.4	13.1	12.4	112.5
	TWSC	516	14.6	5.0	14.4	34.3	99.1
	Signalized	504	42.8	3.4	21.1	7.9	49.3
Movement	LT	588	9.6	3.5	13.9	36.2	144.9
	TH	588	46.4	0.7	19.2	1.4	41.3
	RT	588	8.2	5.0	14.9	60.8	181.7
Total		1764	21.4	3.0	16.1	14.2	75.4

Table 7.6 Descriptive Statistics for Data collected with PTZ

Characteristic		Number of Observations	Average Count	Mean Error	Standard Error	Relative Mean Error (%)	Relative Standard Error (%)
Interval Number	1	77	36.3	6.6	16.4	18.1	45.3
	2	77	40.4	7.2	17.6	17.8	43.5
	3	77	42.7	6.8	23.0	15.9	53.8
	4	77	40.4	6.3	19.6	15.7	48.6
	5	77	37.8	6.8	15.9	17.9	42.2
	6	77	39.8	7.4	17.1	18.5	42.9
	7	77	40.5	7.9	20.2	19.4	49.8
Light	Night	77	45.9	19.4	29.7	42.2	64.6
	Overcast	462	38.6	4.9	16.1	12.7	41.7
Precipitation	None	154	46.5	10.9	22.8	23.3	49.0
	Rain	154	15.2	2.8	5.8	18.5	38.2
	Snow	231	51.4	7.2	21.1	14.0	41.0
Movement	LT	196	25.6	8.8	15.1	34.5	59.1
	RT	147	32.8	-0.6	17.0	-1.7	52.0
	TH	196	59.0	10.8	22.6	18.3	38.4
Total		539	39.7	7.0	18.7	17.6	47.1

7.4.2 Linear Regression Results

Chapter 7.4 introduced two regression models that were used to analyze the affect that local conditions and environment had on turning movement estimation. Equation 7.10 showed the model used for data collected with the Mobile Traffic Laboratory, and Equation 7.11 with the pan/tilt/zoom camera. It must be noted that the predictor variable was the absolute value of the error ($|CE|$). Tables 7.7 and 7.8 show the results of each model, respectively.

The model as described by Equation 7.10 and seen in Table 7.7 represents a set of base conditions where the results could then be compared. These include: camera height of 35 feet, use of one camera, overcast light condition, no camera motion, all-way stop control, and through movement. The model was considered significant with an R^2 of 0.1079. It had an average absolute error of 12.9039 veh/15-min. for the base conditions (intercept).

Table 7.7 shows that increasing the height of the camera to 45 feet decreased the average absolute error by 8 veh/15-min. This was expected because there should be less occlusion in video images with greater heights. It also showed that using two cameras to collect data increased the error by 6 veh/15-min. It was expected that the use of two cameras would improve detection. Explanation of this result could be that in this research two cameras were used only in situations when one camera's video image could not encompass an intersection. Hence, two cameras were used only at large intersections, where there are more lanes of traffic providing greater probability of occluding vehicles.

The effects of light conditions were unforeseen, but they have little significance to the model. However, the results indicate that estimation error increases by a value of 2 veh/15-min. for moderate wind, and a value of 3 veh/15-min. for heavy wind. This result is as anticipated. In addition, the model shows that left and right turn movements are estimated with an error 2 veh/15-min. lower than through movement. It should be kept in mind though, that the average count of left and right turns is roughly half of the average through count. Finally, the model showed that number of lanes, autos, trucks, and pedestrians are considered insignificant in estimation error.

Table 7.7 Regression results for Equation 7.10

$ CE = \beta_0 + \beta_1\text{Hei} + \beta_2\text{Cam} + \beta_3\text{Lmid} + \beta_4\text{Leve} + \beta_5\text{Wmod} + \beta_6\text{Wheav} + \beta_7\text{Twsc}$ $+ \beta_8\text{Sig} + \beta_9\text{Le} + \beta_{10}\text{Ri} + \beta_{11}\text{La} + \beta_{12}\text{Auto} + \beta_{13}\text{Sing} + \beta_{14}\text{Truck} + \beta_{15}\text{Ped}$ $R^2 = 0.1079$					
Variable	β Estimate	Significance	Variable	β Estimate	Significance
Intercept	12.9039	<.0001	Sig	-0.2039	0.9042
Hei	-7.9199	<.0001	Le	-1.9172	0.0432
Cam	6.2588	<.0001	Ri	-1.1810	0.2173
Lmid	-2.3680	0.1178	La	0.0695	0.8964
Leve	1.9706	0.1508	Auto	0.0033	0.8434
Wmod	1.8024	0.0449	Sing	0.1833	0.6883
Weve	3.1622	0.0177	Truck	0.2347	0.5558
Twsc	0.3381	0.8659	Ped	0.1445	0.5650

The model as described by Equation 7.11 and seen in Table 7.8 represents a set of base conditions that include: overcast light condition, through movement, and no precipitation. The model was considered significant with an R^2 of 0.3392. It indicates an average absolute error of 4.3323 veh/15-min. for base conditions (intercept) Table 7.8 shows that night conditions greatly increase the average error by 13 veh/15-min., as expected. This occurs from Autoscoptes' inherent lack of accurate night detection. Again, the model showed that left and right turn movements decrease estimation error by 2, but was considered insignificant for the model. It seems that rain has no effect on the average error as seen by its significance. On the other hand, snow greatly increases the average error by 6 veh/15-min. Also, the presence of autos and pedestrians seem to have a minor increase in estimation error. This is expected for pedestrians due to false detections. Although, the result of autos on estimation error indicates that there is possibility of homoscedasticity of the Flow Conservation method as explained in Chapter 7.2.3. Once

more, trucks seem to have no significance to the overall estimation error of the turning counts.

In both cases of data collection there was a noticeable trend in the performance of estimating turning volumes over time. It seemed that there was a tendency for the performance to deteriorate. This trend could be seen in both the descriptive statistics and regression models developed. Accordingly, this trend could not be clearly confirmed with statistical analysis. It should be noted that this phenomena could be explained with weather and light conditions that fluctuate over time. Hence, an initial detector layout made for a certain light and weather conditions, can be less optimal if light and weather conditions were to drastically change.

Table 7.8 Regression results for Equation 7.11

$CE = \beta_0 + \beta_1 Ni + \beta_2 Le + \beta_3 Ri + \beta_4 Rain + \beta_5 Snow + \beta_6 Auto + \beta_7 Sing + \beta_8 Truck + \beta_9 Ped$ $R^2 = 0.3279$		
Variable	β Estimate	Significance
Intercept	4.3323	0.0410
Ni	13.3619	<.0001
Le	-2.0296	0.1294
Ri	-1.9739	0.1690
Rain	-1.3374	0.4618
Snow	5.6893	0.0002
Auto	0.1363	<.0001
Sing	-0.3228	0.3408
Truck	-0.1927	0.8884
Ped	0.1404	0.0942

6.5 Closure

The evaluation results indicate that using Autoscope together with the flow conservation method is feasible. It should be kept in mind though, that it is challenging to locate spot detectors that are passed by all vehicles of a certain flow. In addition, the quality of detector counts is critical for the quality of the turning flow estimates due to rather small data redundancy present in the problem.

The limitation of the proposed technique is where the number of spots with sufficiently diversified set of turning flows is too small to extract all the turning flows. Intersections with single-lane approaches are an example. On the other hand, intersections with exclusive lanes for either left or right-turning flows can enhance the proposed estimation method.

The camera's elevation along with intersection size is critical for estimation quality due to the occlusion phenomenon. Larger intersections with multiple lanes can increase this effect, but as the results showed it can be reduced with a higher camera elevation. Also, motion of the cameras due to wind may cause numerous false counts. Placing a detector on a solid color background can reduce this effect. In addition, the detectors should be located where no stopped vehicles are expected. The combination of a vehicle stopped in a detection spot and a camera motion can cause multiple detections of the same vehicle. The error collected from the detectors is over emphasized resulting in poor flow estimation.

The light conditions experienced during the midday and evening have little effect, but it was verified that estimation errors greatly increase during the night. As for

inclement weather conditions, rain has little to no effect. However, snow conditions can increase detection error.

Although the findings did not tell us how many cameras should be used, we feel that for larger intersections it is better to use two cameras if possible and focus on portions of the intersection instead of using one camera with a wide-angle lens. Objects far from the camera become smaller, and undersized detectors have to be placed there at the expense of the counting quality.

In conclusion, the proposed method of extracting turning flows from multiple detector counts is valid and not necessarily associated with video detection. It can be combined with any detection technique that allows fast setting of multiple detectors with localized detection spots. Micro detectors placed on pavement and retrieved after counting can be used instead. Today's technology allows for building such small devices with their own power source and data storage capabilities.

8. VIDEOTRAK METHOD AND EVALUATION

VideoTrak is considered a tracking detection system. Tracking detection systems identify individual vehicles in an image and track the vehicles through that image. It does so by determining the location of a vehicle from pixel intensity changes that occur from frame to frame in a video image. Groups or “blobs” of changing pixels represent moving vehicles. VideoTrak uses tracking strips to define areas for tracking, and then similar to Autoscope has detection zones to retrieve relevant data. Please refer to Chapter 4 for more explanation of the VideoTrak video detection system.

This chapter explains the method and evaluation procedures used to analyze data using the VideoTrak detection system. The VideoTrak system was selected for its tracking capabilities and the ability to acquire per-vehicle-record output with a special DOS program, called Academia. A tracking strip method was developed especially for the VideoTrak system. This method uses tracking strip per-vehicle-record output to define turning movements at intersections. A Visual Basic program was developed to interpret the data and provide counts of turning vehicles including vehicle type classification.

This chapter first introduces the Academia version of VideoTrak. A detailed explanation of the programs’ functions and operations is given first. Then, the tracking strip method for determining turning movements from tracking strip output is explained, along with the explanation of the algorithm used. Subsequently, the methods used to

extract the raw data using the Academia version of VideoTrak are described. Finally, there is an explanation of the statistics used to analyze the data, and conclusions made from the results.

8.1 Academia Version of VideoTrak

The Academia version of VideoTrak is a DOS program that is able to acquire per-vehicle-record data of every vehicle within a tracking strip three times a second. It originally was a testing program that Peek Traffic Systems used to test their products. The program must run in DOS, and when the normal Windows version of the VideoTrak software is closed. It reports 16 columns of data every third of a second. Below are descriptions of each column of data.

- Tap – This value can either be 0 or 1. It is a flag that identifies when the spacebar on the keyboard is pushed when the DOS program is running;
- Veh ID – This is a random number between 0 and 31. This number is assigned to a vehicle when it enters a tracking strip;
- Strip – This is a number between 0 and 4. The number identifies the tracking strip within the field-of-view;
- Track – This value represents the object age within the tracking strip. It cannot have a value less than 300 milliseconds. Hence, vehicles aren't tracked until it is detected for at least 300 milliseconds.
- S_x – This value represents the starting x-coordinate of the front edge of an object in pixels.

- S_y – This value represents the ending y-coordinate of the front edge of an object in pixels.
- E_x – This value represents the starting x-coordinate of the end edge of an object in pixels.
- E_y – This value represents the ending y-coordinate of the end edge of an object in pixels.
- Dist – This value is the estimated distance from the object to the camera in feet.
- Length – This value is the apparent visual length of an object in feet.
- Speed – This value is the estimated speed of an object in mph.
- L_{unc} – This value is between 1 and 5, and represents the uncertainty of the Length value. Where a value of 1 is very certain and 5 is not certain.
- S_{unc} – This value is between 1 and 5, and represents the uncertainty of the Speed value. Where a value of 1 is very certain and 5 not certain.
- Zones – This value is a 32-bit binary number that represents the zone being occupied by an object.
- Z_{prev} – This value is a 32-bit binary number that represents the zone previously occupied by an object.
- WW - This value can either be 0 or 1. It is a flag that identifies when an object is moving the wrong way through a tracking strip.

The most important column of data in the per-vehicle-record output is the X and Y coordinates. The coordinates help determine the position of a vehicle on the video image. However, VideoTrak does not give true positions of vehicles. Instead, the position is related to the direction and orientation of the tracking strip, and is given with the accuracy of strip width (vehicle position is represented by a point of the strip's centerline).

A tracking strip is defined by a closed polyline with a maximum of eight vertices. Tracking within a strip can be defined in either a horizontal or vertical direction. An example of a tracking strip is shown in Figure 8.1. The figure shows horizontal lines, representing pixel rows, which renders tracking in the vertical direction. The solid figures represent true positions of a vehicle traveling through the tracking strip, whereas the dashed figures represent the positions of that vehicle as procured through the VideoTrak tracking algorithm. As shown in the figure, per-vehicle-record for this tracking strip would provide an accurate y-coordinate because the tracking strip is tracking vertically. However, the x-coordinate of the PVR reflects the horizontal centroid, rather than its true x-coordinate. This is similar for a tracking strip that horizontally tracks vehicles. The per-vehicle-record would indicate a true x-coordinate, and an inaccurate y-coordinate depending on the orientation of the tracking strip. Hence, per-vehicle-record position data is highly dependent on the direction and orientation of the tracking strip, and should really be considered only one-dimensional tracking, either horizontal or vertical.

coordinates along with other data including Tap, Veh ID, Strip, Track and Length, which are used in our method. Figure 8.2 also shows that the tracking strip is split into two zones, a right-turn zone and a through movement zone. Only vehicles that enter the tracking strip close to the beginning of the strip and moves in the expected direction are considered valid, others are ignored as invalid. Then, as a vehicle exits the tracking strip, the point of departure determines the turning movement for that vehicle. Therefore, if a vehicle enters the strip at the strip beginning and exits being in the right-turn zone, this vehicle is considered a right-turn movement.

A post-processing program supplemented to Academia was developed to execute the tracking strip method concept. The Visual Basic software was chosen to develop a user-friendly graphic interface that could be used with any Academia output. The next section describes the algorithm for the program and how it was developed.

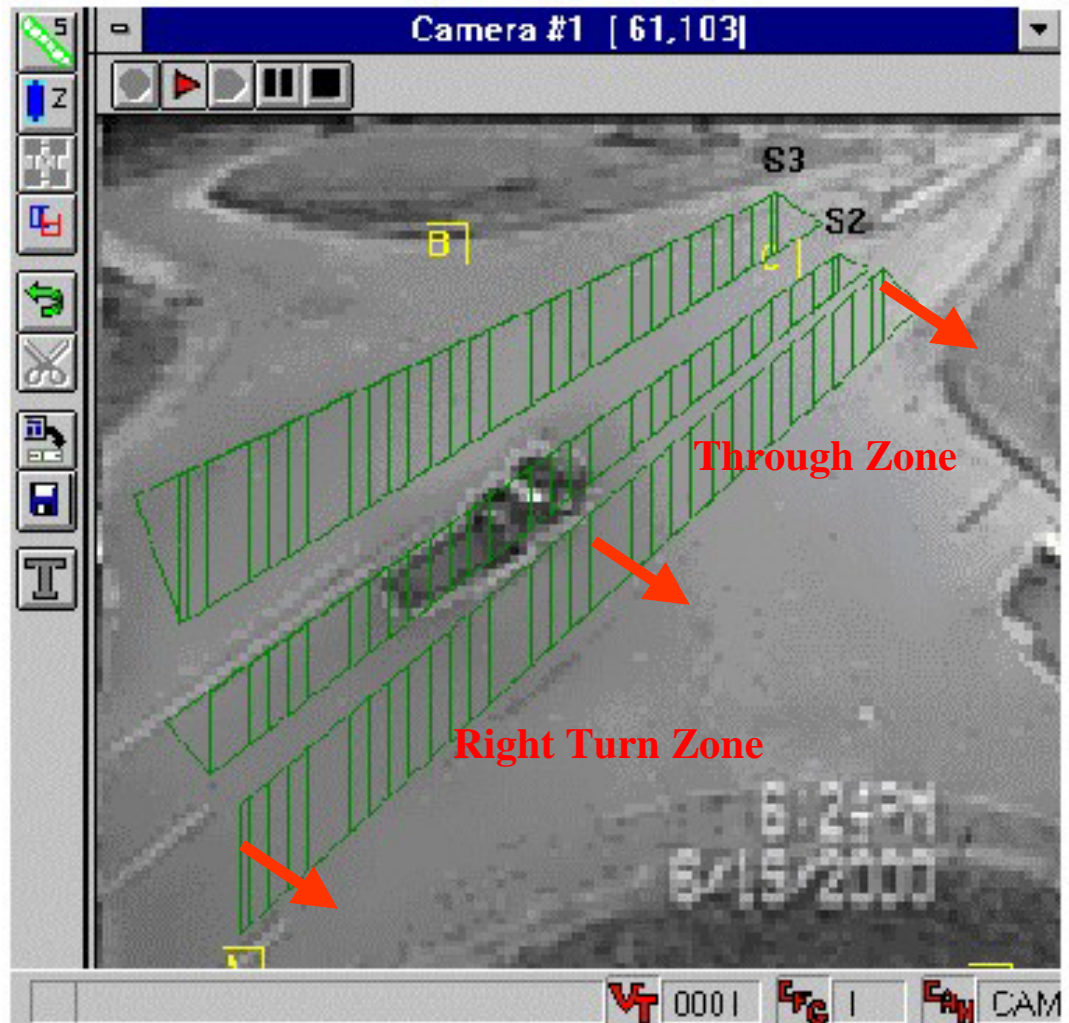


Figure 8.2 Example of tracking strip method

8.2 Tracking Strip Algorithm

As Section 8.1 explains, output of the Academia program is provided in row data, where each row represents a per-vehicle-record. The method used only 8 of the 16 pieces of column data available. These are the Tap, Veh ID, Strip, Track, S_x , S_y , E_x , E_y , and Length fields. The Tap field was utilized as a time reference to correlate the data collected with the ground truth data. This was the only available way of doing so. The Tap field is similar to a counter, in that whenever the space bar on the keyboard was punched, it increased the counter by one. The Tap field always began at zero and increased by one until punched 10 times. After which it read the value of 1 when hit 10 more times, and then the value of 2 when hit 10 more times, and so on. It seemed the Tap field could only hold one bit of information. However, it was fortunate that data was collected in 15-minute intervals for each two-hour tape, and the Tap field was only needed to be struck up to 9 times.

The Veh ID and Strip fields were used to differentiate flows and vehicles within those flows. The Strip field defined what flow was being tracked, and the Veh ID was used to identify individual vehicles that traveled within the strip. The method looked for similar Veh ID's to create a tracking history of a vehicle as it moved through the strip. The tracking history of a vehicle was defined with the Veh ID's Track data. The Track information was the time measurement as a vehicle moved within a strip. For example, when a vehicle entered the strip it did not start tracking until at least three hundredths of a second had passed. Once being tracked the vehicles' Track time increased until it left the strip. However, the only way to logically know when the vehicle left the strip was when

another vehicle with the same Veh ID entered some other strip, hence, beginning a new Track value from three hundredths of a second. Therefore, the logic used to define same vehicles was if the Track age of the previous record was smaller than the Track age of the current record. If it was not, then it must be a new vehicle.

The S_x , S_y , E_x , and E_y fields were used to recognize the position of vehicle inside the tracking strip. The S_x and S_y represented the pixel coordinates of the starting edge of a vehicle, and the E_x and E_y represented the ending edge. In our method we used starting edge of a vehicle to determine the tracking history and when it had left a strip, and therefore could define a turning movement for that vehicle. Although, it was determined through our research that the S_x and S_y terms did not always define the starting edge of a vehicle, rather it depended on how the vehicle was tracked within a video image. Figure 8.3 shows how pixels are defined for a video image for the VideoTrak system, and the four directions for which a vehicle can be tracked. For vehicles being tracked up and left the start pixels values reported by the Academia Program were correct in reporting the starting edge of a vehicle. However, for vehicles being tracked right and down, the start pixels reported actually defined the ending edge of a vehicle. Therefore, the final program considered these attributes when defining turning movements.

Finally, the Length field was used to calculate overall lengths of the vehicles, and then classify the vehicles according to specified lengths. As vehicles traveled through the strip, the reported lengths for every point in the tracking history were summed up and then averaged.

The tracking strip method must use information of the tracking strip as input into the algorithm as show in Figure 8.4. This includes: the number of strips to be analyzed, if the

strips are horizontal or vertical, if the strips are tracking right/down or left/right, the pixel range that define the entrance to the strip, the pixel range to define right turn movement, the pixel range to define left turn movement, the pixel range to define through movement, and the pixel range used to classify vehicles.

The pixel groups that define movements are self-explanatory since it is the concept of the method. However, there needs to be explanation for the pixel groups used to classify vehicles. When a vehicle was first entering a strip to be tracked, the entire vehicle did not have to be inside the strip. Hence, the length reported was not the entire length of the vehicle. This was the same for vehicles that exited the strip. Therefore, instead of classifying vehicles by averaging their lengths across the entire strip, pixel groups were defined where a more accurate calculated length could be made. There were three classification types available, specified by length parameter that was set inside the program by the user.

After the tracking strip information is inputted into the program, it uses the algorithm in Figure 8.4 to define turning movements. First it opened a database that stored the output given by the Academia program. A database had to be used because an average 2-hour tape could generate around 150,000 lines of row data. While reading the database data record, it first asked if the data record for that VehID had entrance pixels within the defined entrance pixel group of that strip. If not, it would read the next data record. If it did, it would then ask if it is the first time that vehID had been used. If it was, it updated the data record information for that particular Veh ID and read the next record. If not, it would then ask if it was a new vehicle for that VehID. This was done through the logic if the current Track time was less than the previous Track time for a particular

Veh ID and strip. If so, it would then classify the vehicle by length and movement, and begin a new data record for the current Veh ID data. If not, it would update the data record with the current Veh ID data and ask if it was the end of file. If it were not the end of file it would read the next line of data and go through the algorithm again. If it were the end of the file it would classify the remaining Veh ID data records, and write the results to a file. Appendix A gives a printout of the code used in the Visual Basic program.

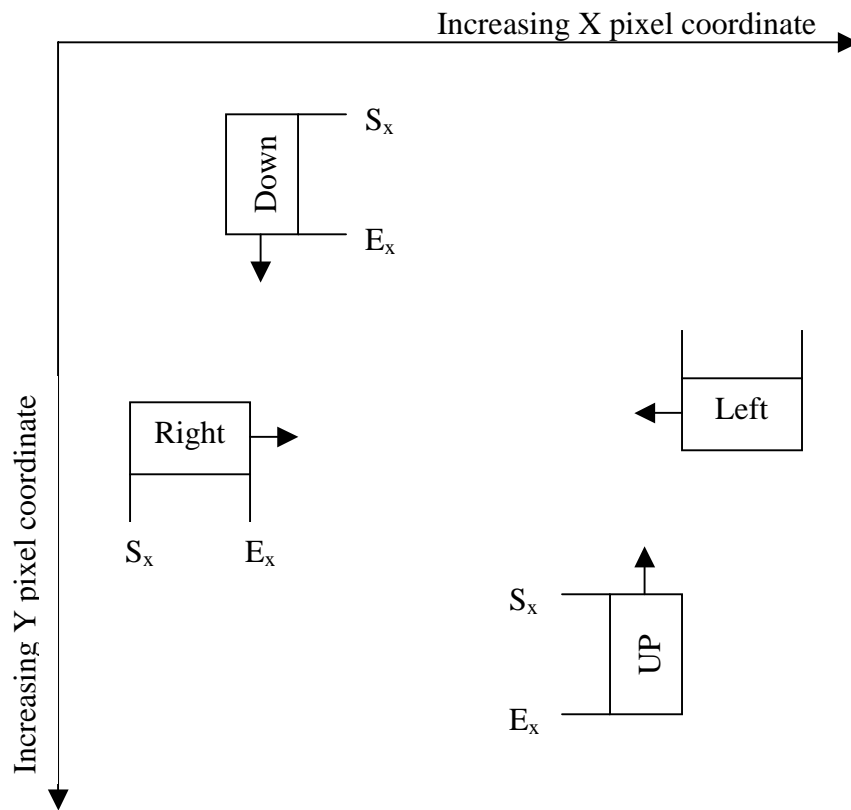


Figure 8.3 Reported Start and End pixels for VideoTrak



Figure 8.4 Algorithm for Tracking Method Program

8.3 Data Extraction

The Academia program was used to retrieve data from three videotapes. The VideoTrak program only allows for placement of five tracking strips per video image. To extract data for twelve movements, each two-hour tape had to be analyzed three or more times. Time constraints did not allow to process more tapes than three, one for each intersection shown in Figure 6.5. Tapes with no inclement weather or light conditions were chosen (base conditions).

Tracking strips were placed in individual traffic lanes and stretched across intersections as seen in Figure 8.5. Since tracking strips were assigned to lanes, the maneuvers classified by a single strip were the same maneuvers that used the lane. After the tracking strips were placed, observers watched traffic of vehicles within individual strips to define turning movement and classification zones for the strips.

After the tracking strips were placed and turning movement and classification zones were defined, the Academia program was used to collect the data. During the data collection, every fifteen minutes an observer would hit the space bar in order to mark the data aggregation intervals. Once the Academia program was executed, the data saved in the Excel format was imported to a Microsoft Access database where it could be further processed using the developed Visual Basic program.

After running the program and obtaining estimated fifteen-minute turning counts, they were matched with the fifteen-minute ground truth count for analysis. The same measures of performance were used as in the Autoscope evaluation.

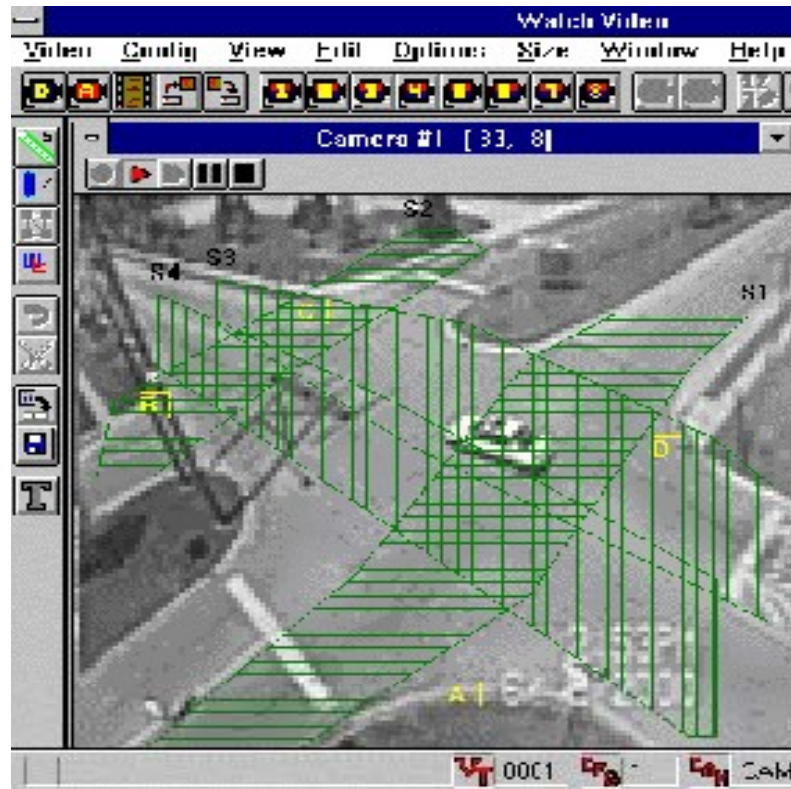


Figure 8.5 Example of Tracking Strip Layout

8.4 Method Evaluation

Two sets of statistics were developed for the evaluation of the VideoTrak detection system. The first set included turning movement estimation by movement only, whereas the second set described turning movement estimation by movement and vehicle type. Figure 8.1 and Table 8.1 shows the results of the estimation.

Table 8.1 Results of VideoTrak detection system

	Tape	Intersection	No. Obs.	Avg. Count (veh)	Mean Error (veh)	Std. Error	Rel. Mean Error	Rel. Std Error
VideoTrak (No Vehicle Classification)	100	AWSC	84	15.21	-1.20	5.33	-7.90%	35.04%
	109	TWSC	84	12.18	0.93	5.66	7.62%	46.50%
	133	SC	77	47.17	0.53	7.15	1.13%	15.16%
Total			245	24.22	0.07	6.07	0.30%	25.06%
VideoTrak (Vehicle Classification)	100	AWSC	252	5.07	-0.40	3.36	-7.90%	66.25%
	109	TWSC	252	4.06	0.31	3.04	7.62%	74.87%
	133	SC	231	15.72	0.18	14.07	1.13%	89.52%
Total			735	8.07	0.02	8.32	0.30%	103.13%

The results for count estimation by movement only show that the mean absolute estimation error is around +/-1.00 veh/15-min. for all cases, with a minimal value of 0.07 veh/15-min. considering the errors in relation to the entire sample. The standard error was 6.07 veh/15-min. for the entire sample. Relative to the average number of vehicles actually counted, one can conclude that the estimate was not as accurate as could be seen by looking at the absolute values only. Relative mean error describes the error as a percentage of the volume of traffic counted. It showed 0.30% error with relative standard error of 25.06%. For the all-way stop intersection there was an overall underestimation, and a small overestimation for the two-way stop control and signalized intersections. The standard errors for all cases were reasonably small with the highest standard error having a value of approximately 7. The all-way stop and two-way stop intersection had approximately -8% and +8% mean error, respectively, and the signalized intersection

only had approximately +1% mean error. The relative standard errors for the all-way stop and two-way stop intersection were +35% and +46%, respectively, and the standard error of the signalized intersection was approximately 15%

It seems that estimation for the signalized intersection is much better than for the all-way and two-way stop intersections, at least in the small sample collected. This could be attributed to the fact that the all-way and two-way stop intersections had more tracking strips with multiple movements than the signalized intersection. The all-way stop controlled intersection had two strips with three movements, two strips with two movements, and another two with single movements. The two-way stop had three strips tracking two movements and eight strips tracking single movements. The signalized intersection had only one strip tracking two movements, while ten strips tracking single movements.

The results for count estimation by movement and classification show an overall decrease in mean and standard errors. This is due to the aggregation of data for a larger number of observations. The mean error was less than +/-1 veh/15-min. for all the cases. The standard error was approximately 3 veh/15min. for the all-way and two-way stop intersections, and 14 veh/15-min. for the signalized intersection. The relative mean error is the same as it should be. However, the relative standard errors approximately doubled for all-way stop and two-way stop intersections, and were six times larger for the signalized intersection as compared to statistics made just by movement. Therefore, there is a great error in classifying vehicles using the methods introduced in this chapter and seen in Figure 8.1. This can be attributed to three things. The first is that classification for the ground truth vehicles was done by interpreting their weights as described in Chapter

5.4. Alternatively, video detection can only classify vehicles according to their lengths. In addition, there was a need for individual strip specific calibration of critical vehicle lengths for classification. Only one strip per field-of-view was used to calibrate vehicle lengths. For some reason, the calibration of that strip did not agree for the others, resulting in classification errors. Finally, the other possibility could be that during the calibration process of the video images, inaccurate distance and height measurements were entered into the VideoTrak program. But the researchers feel this is highly unlikely.

In summary, the VideoTrak system outperformed the Autoscope system for accurately estimating turning movement counts at intersections. However, a limitation of the VideoTrak system method is the extensive time needed to implement the method. Contrast to what was initially thought, the VideoTrak system can only perform one-dimensional tracking, thereby creating more difficulty to develop a more feasible concept of turning movement estimation. We emphasize that these statements in no way reflect the potential capability of VideoTraks' standard features to detect vehicles and obtain relevant traffic data.

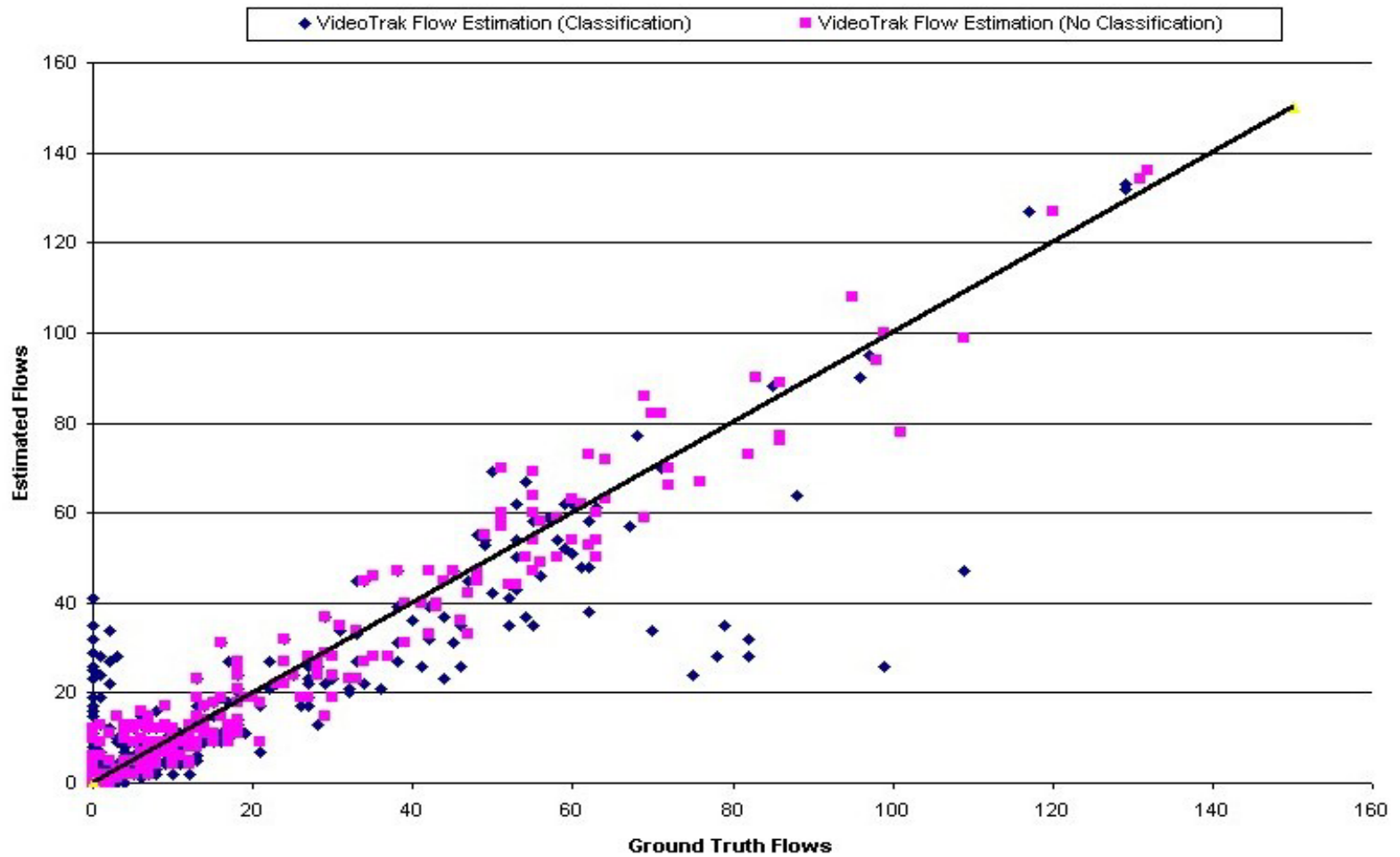


Figure 8.1 VideoTrak estimation with and without classification

9. COMPARATIVE EVALUATION

The comparative evaluation of both the selected systems considered not only their performance, but also the implementation issues. After careful consideration, we conclude that Autoscope is the system that should be used if a prototype portable video detection system is to be constructed today. It would combine a time efficient technique of data collection with reasonableness of the results. This chapter gives the details of the comparative evaluation. The next chapter gives specifications for a portable video detection system.

9.1 Implementation Issues

Implementation of the Autoscope-based method as discussed in Chapter 7 is straightforward. The Autoscope video detection system allows for easy placement of spot detectors on a video image. After obtaining data from the detectors the post-processing method is simple and fast. In addition, approximately one-half to a full hour of post-processing is needed to obtain turning movements for an intersection. This does not include the time it would take to extract data from the field; this would incur double the time of the videotaped data collected.

Conversely, the implementation of the VideoTrak method as discussed in Chapter 8 is time consuming and laborious. Similar to Autoscope, the placement of the tracking strips within the video image would be effortless. However, since the Academia version of VideoTrak allows only placement of five tracking strips per video image, it would quadruple the time necessary to extract the per vehicle record data from the field. This includes the time it would take to collect the video images. In addition, the post-processing method indicated in this research would approximately take one to two hours of time to perform. Therefore, in consideration that this research is trying to minimize data collection time at intersections, the Autoscope system does better.

This conclusion summarizes the current version of the systems. The difference in the post-processing time can be eliminated if the manufacturers of the VideoTrak system expand their software to include up to twelve tracking strips, and improve the operation of the Academia software.

9.2 Performance Measures

The individual evaluations of both the Autoscope and VideoTrak used counting error as their measure of performance. Autoscope had an average absolute error of 4.0 vehicles, and a standard error of 16.8 vehicles. Furthermore, relative to the average vehicle count, Autoscope overestimated turning movements by 15.4% with a relative standard error of 65.3%. Conversely, the VideoTrak had an average absolute error of only 0.07 vehicles with standard error of 6.07. Additionally, it had a relative mean error of

only 0.30% with a relative standard error of 25.06%. The numbers prove that the VideoTrak unit significantly outperformed the Autoscope system. Figure 9.1 also illustrates the comparison of the two systems. The figure clearly indicates better estimation performance using the VideoTrak detection system. As for classification, there is no way to classify vehicles using the Autoscope detection system. On the other hand, VideoTrak can classify vehicles, but with noticeable inconsistencies at least with the proposed method. Therefore, in terms of accurate turning movement estimation VideoTrak is the better detection system.

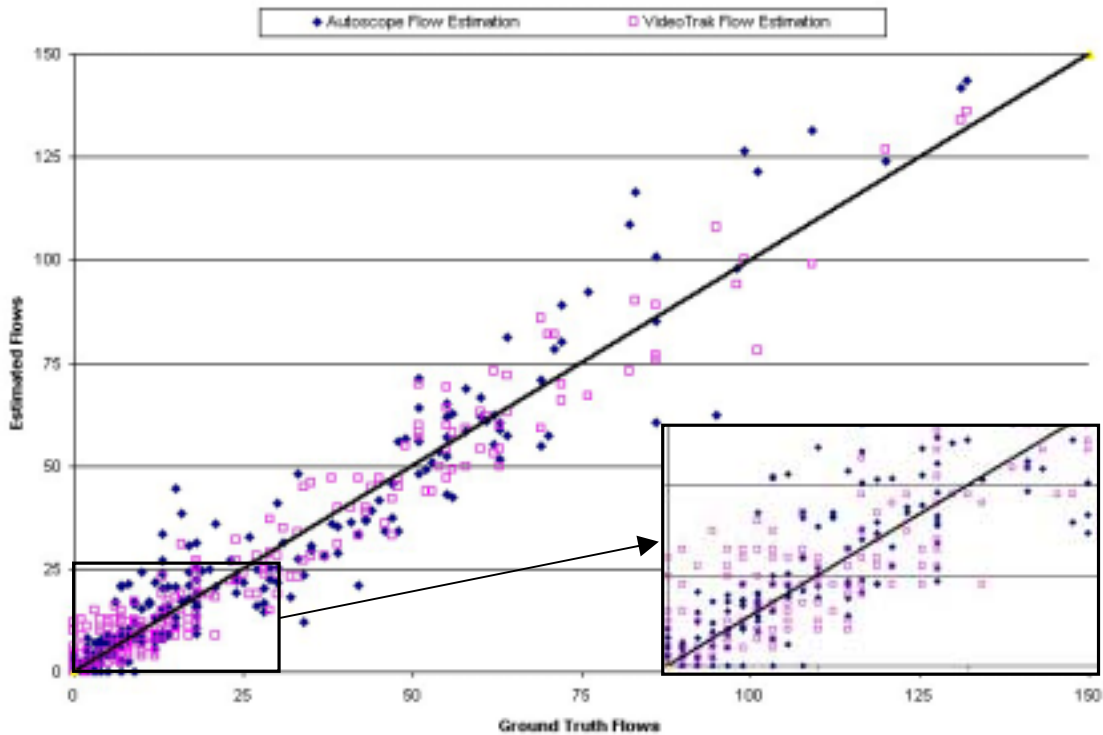


Figure 9.1 Comparative Evaluations of Autoscope and VideoTrak

9.3 Strengths and Weaknesses

Foremost, the Autoscope detection system is a trip-wire system as mentioned in Chapter 5.2.1. The advantage of the Autoscope video detection system is that there is flexibility in placement of both the number and direction of detectors. Remember, Autoscope allows the placement of the 99 possible detectors. In addition, the direction for a group of pixels of a detector is not important, only pixel intensity change is needed for detection. Furthermore, the setup of the detector layouts are generally much easier, and limited processing power is required since a small group of pixels is observed.

The disadvantages of the Autoscope video detection system are that it is very susceptible to detection error from local and environmental conditions. These can include shadows, precipitation, camera motion, and nighttime. Therefore, unless the system is collecting data from a 'perfect' image, the Autoscope can create much false detection as seen in this research.

The advantages of the VideoTrak system is that the same local and environmental conditions as described above can be handled since it uses the blobifying affect to track vehicles as described in Chapter 5.2.2. In addition, the tracking algorithms used by the VideoTrak system provide excellent detection of individual vehicles. Therefore, the system is able to provide more accurate and reliable results.

However, disadvantages include that it can only precisely track one-dimensionally, either horizontally or vertically. Hence, the video image must be positioned towards individual approaches of an intersection, rather than encompassing the entire intersection itself. This is because the tracking strips of the VideoTrak system are lane specific, and cannot be

drawn across lanes. In addition there is a limit to the capability of the tracking strip in that only five can be used for one video image, and the tracking strips can only track up to thirty-two vehicles within a video image. Finally, the VideoTrak system must require more variable inputs along with processing power to track vehicles.

10. SYSTEM SPECIFICATIONS

This document provides specifications of a portable video detection system sufficient to build a prototype unit. The specifications are the result of a collaborative effort between Purdue University and the Indiana Department of Transportation (INDOT) through the Joint Transportation Research Program (JTRP). This project is identified as Indiana SPR-2394 titled *Portable System for Collecting Intersection Data*.

The *Portable Video Detection System* is intended to introduce two new features to the video detection technique: counting turning vehicles at intersections and video detection system portability. The system is designed to be easily relocated and set at various locations including intersections, freeways, bridges, and roads. Specifically, the system has two functions: (1) acquire traffic video images; (2) extract traffic data from the acquired video images. The system use is not restricted to these functions. It is designed as a modular system that allows the integration of other components and functions for features for traffic detection, surveillance, and monitoring.

These specifications describe the components, equipment, operational characteristics, and installation requirements for the prototype of an operational *Portable Video Detection System*. As such, these specifications are geared to defining items that are unique to the system. Where these specifications describe portions of the system in general terms, but not in complete detail, only the best general practice is to prevail, and only materials and workmanship of the best quality should be used.

10.1 System Design and Operation

This section introduces the components of the *Portable Video Detection System* with explanation of how these components interconnect and function together. In addition, there are explanations of two alternative methods of operation during implementation that demonstrates the robustness of the *Portable Video Detection System*.

The system is divided into two categories of components, exterior and interior. The exterior components include the necessary equipment to obtain and record video images in the field. The interior components include the devices required to analyze and retrieve data from the recorded images.

10.1.1 Exterior Components

The exterior components shall be used during outdoor field operation of the *Portable Video Detection System*, and is comprised of: the system structure, data acquisition system, data storage and processing system, and the power supply system. The integration of all these components shall perform the observation and recording of all movements associated with pedestrians, bicycles, and vehicles at an intersection. The fully integrated system should operate unattended for a minimum of sixteen hours, and should require no more than one person to setup and remove the system.

10.1.1.1 System Structure

The system structure should consist of a portable trailer unit with an attached mast (Figure 10.1). In addition, a housing container should be affixed to the trailer to house the electronic equipment necessary for the system. The portable trailer unit is used as an instrument to haul the exterior components to a pre-specified field site where data can be collected. The mast is used as a medium to raise cameras to a desired elevation to obtain video images without excessive occlusion of vehicles. The housing is used to shelter and protect the electronic equipment and other equipment necessary for the system. These should include the data acquisition system, data storage and processor system, and power supply system.

10.1.1.2 Data Acquisition System

The data acquisition system should consist of the necessary parts to acquire video images for digital storage (Figure 10.2). This should include two digital video cameras, two environmental camera housings, a pan/tilt camera mount attachment, video and power cabling, and camera controller. The function of this system is to provide high-quality images to the data storage and processing system. It is recommended that a pre-assembled data acquisition system with similar product brands be acquired (i.e. Pelco or Panasonic products)

10.1.1.3 Data Storage and Processor System

The data storage subsystem should include the necessary computer hardware and software needed to store video data digitally, and the necessary hardware and software used in video detection. This includes a transportable computer processor, computer monitor, keyboard, and mouse (Figure 10.2). Its purpose is to collect and save sixteen hours of digital video data, and run necessary software applications including video detection.

10.1.1.4 Power Supply System

The power supply system may consist of either a rechargeable battery supply or gas/diesel fuel generator that is able to power all equipment for a period of twenty-four hours. The battery supply should use either Deep Cycle or Marine batteries with a recharging capability by plugging into any 120/240V electrical outlet. These battery types are specifically designed for prolonged discharging and repetitive recharging. The power supply should include the necessary batteries, wires, inverters, and recharger to supply power to all of the electrical equipment, and should be located within a separate compartment within the housing of the system structure.

A deep cycle battery usually denotes the battery capacity rating in amp-hours. For example, a deep cycle battery with a capacity of 75 amp-hours suggests that the battery

can pull 1 amp continuously for 75 minutes, or 5 amps for 15 hours. The combined ampere consumption of all electrical components should be estimated first to approximate the number of batteries needed for a continuous power supply of 24 hours.

An alternative method of power supply may use a generator powered with gas or diesel fuel. The Indiana Department of Transportation recommends using diesel generators as safer. The generator shall be the kind that is able to supply uninterrupted current at characteristics sufficient for computer and electrical equipment. If normal fuel tank capacity of the generator does not allow the specified operation of twenty-four hours, then an additional fuel container should be provided. The generator with gas/diesel container and any other necessary parts of this power supply system should be located within a separate compartment within the housing of the system structure.

10.1.2 Interior Components

This equipment is to be used indoors for analyzing video images stored in the data acquisition system. These are the selected video detection system and another data storage and processing system. The integration of these two components shall perform the video detection analysis of archived video data as collected from the exterior components of the *Portable Video Detection System*. Such analysis is done in-house where proper video detection analysis techniques employed.

10.1.2.1 Data Storage and Processor System

This system is the same as the one described in section 10.1.1.3. The concept is to have two data storage and processor systems that can alternate the functions of data storage and data processing during operation. For example, when one system is employed by the exterior components it is used just as a data storage unit. When the other system is employed by the interior components, it is used as a data storage and processing unit. Hence, one data storage and processor system can be used to analyze data indoors while the second system is outdoors collecting and recording video data, and vice versa.

10.1.2.2 Video Detection System

The video detection system should be any reliable video detection system that can analyze video images of traffic and report relevant traffic parameters such as volume, speed, and occupancy. However, one should consider the Autoscope 2004 video detection system as recommended.

10.1.3 System Integration

As previously described, the *Portable Video Detection System* is comprised of two component types, exterior and interior. Figure 10.1 illustrates the exterior components of the system. It shows a trailer unit with two housing compartments and an extended mast with a raised camera to obtain video images. This should be the fundamental design for the *Portable Video Detection System*.

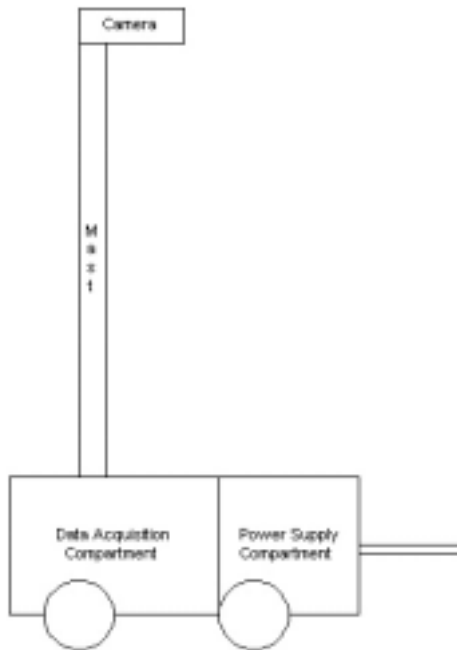


Figure 10.1 Fundamental design

Altogether, the data acquisition, data storage, and power supply systems should be stored in an enclosed housing attached to the trailer. This housing should be partitioned into two compartments. One for the data acquisition and storage system, and the other for

the power supply system. The system structure as well as the camera, camera housing, and cabling of the data acquisition system should be the only equipment exposed to the environment during operation. Figure 10.2 illustrates the connectivity of the data acquisition, data storage, and power supply systems. It shows that cameras with housings and pan/tilt mount attached to the mast transmit video images through cabling to the data storage computer. Furthermore, the power supply system connects to both the data acquisition and storage systems to provide the necessary power to run the electrical equipment. Ultimately, the data storage system should be joined with a video detection system in house to analyze stored video images and collect traffic data as shown in Figure 10.3.

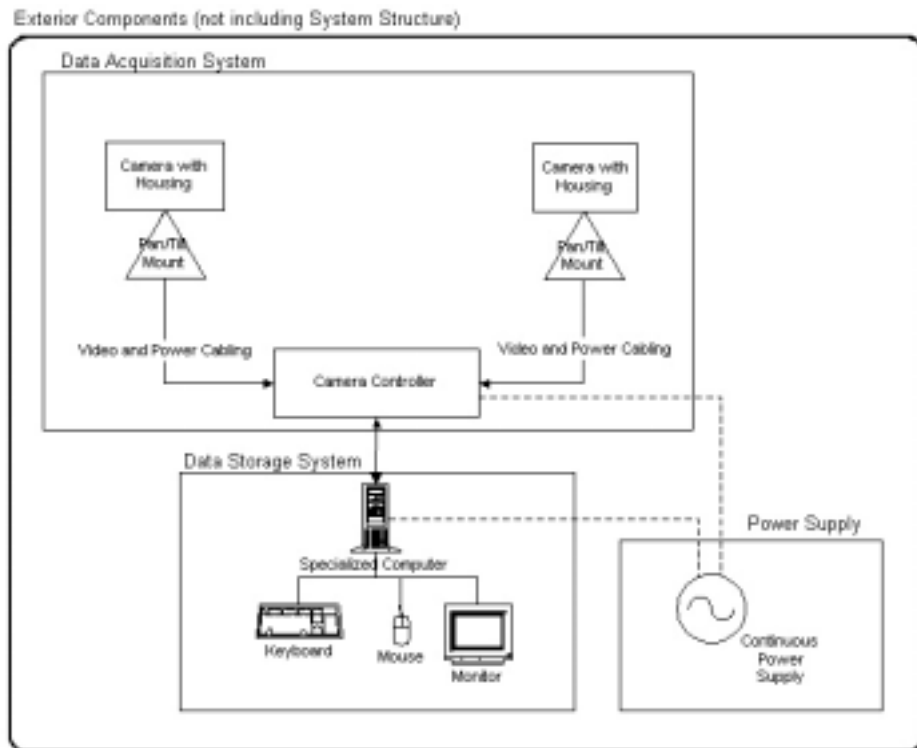


Figure 10.2 Exterior Component Connectivity

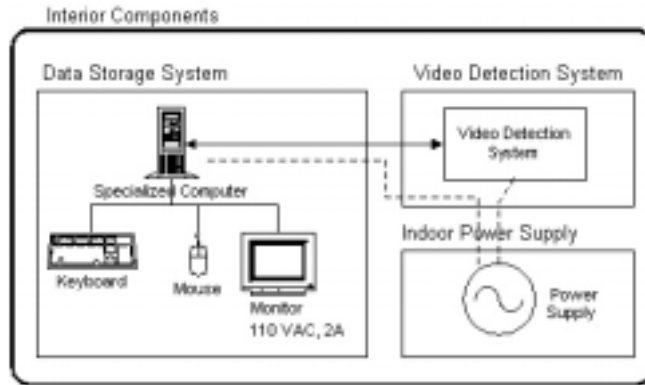


Figure 10.3 Interior Component Connectivity

10.1.4 System Operations

The robust design of the *Portable Video Detection System* allows for two methods of operation. The system should be able to collect intersection traffic data either with offline processing of the video images, or real-time field processing. This section will describe both operations and explain their benefits and limitations.

10.1.4.1 Offline Processing

Off-line processing of the *Portable Video Detection System* should occur when video images are collected in the field, and processed in house using a video detection system. The complete process is shown in Figure 10.4. First, a real-world site should be

chosen for investigation. This should include preliminary site observations as to acquire the best location to setup the *Portable Video Detection System* exterior components. The trailer and most of the system structure should be setup in such a way as not to obstruct the view of drivers, and the system should not be situated under overhead electrical, phone, and cable wires. After deciding upon the optimal placement of the system for data collection, the two video cameras with camera controller (from the data acquisition system) should be used together to obtain the video images needed for data collection. The video images should be transmitted to the hard disk of the data storage and processing unit where they can be stored for analysis. Once all necessary images are obtained the exterior components should be dismantled in such a way as to store the *Portable Video Detection System* in a safe, secure storage facility. Then, the transportable computer of the data acquisition system can be taken from the exterior components and taken inside to connect directly to a network and video detection system. This is where analysis of the video images should be done.

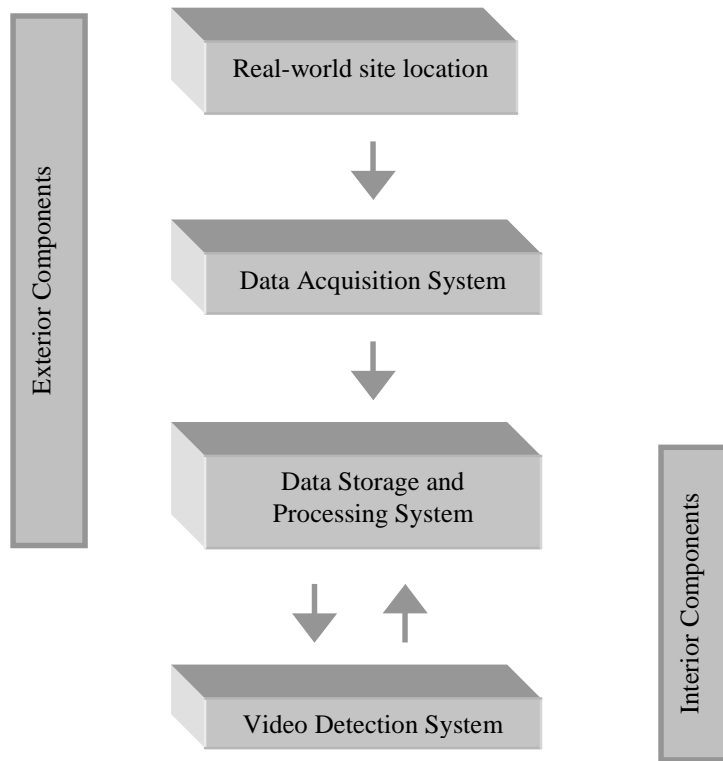


Figure 10.4 Offline Processing Operations

There are both benefits and limitations of operating the exterior components of the *Portable Video Detection System* as a video storage system, rather than a fully self-contained mobile video detection system that provides real-time traffic data. Benefits include having the ability to perform multiple analyses on a single portion of recorded video to collect traffic data, less time needed to set-up the system, the user need not be proficient in the use of video detection equipment, and there is less financial liability when using the *Portable Video Detection System* because the video detection unit would not be housed in the system structure. Limitations include the incapability for collecting

real-time traffic data, and an increase in the amount of total time needed to collect traffic data at intersections and then to process it indoors.

10.1.4.2 Field Processing

Field processing of the *Portable Video Detection System* occurs when video images are collected and analyzed in real-time. The complete process is shown in Figure 10.5. Once more, a real-world site should be first chosen for investigation. This should include preliminary site observations as to acquire the best location to setup the *Portable Video Detection System* exterior components. The trailer and mast of the system structure should be setup in such a way as not to obstruct the view of drivers, and the system should not be situated under overhead electrical, phone, and cable wires. After deciding upon the optimal placement of the system for data collection, the two video cameras with camera controller (from the data acquisition system) are used together to obtain the video images needed for data collection. The video images then are transmitted to the data processing unit where they can be analyzed in real-time, and to the hard disk of the data storage unit where they can be stored for off-line inspection and/or analysis. This would involve a complete setup of the video detection system on-site. Once all necessary data are obtained the exterior components can be dismantled in such a way as to store the *Portable Video Detection System* in a safe, secure storage facility.

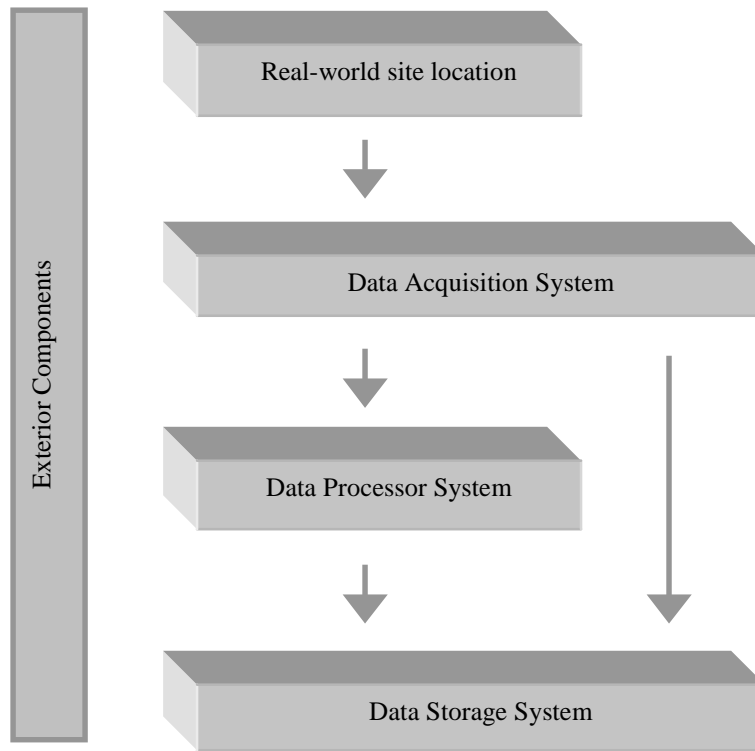


Figure 10.5 Field Processing Operations

The benefit of real-time processing is that data is collected in real-time, which would not require multiple playbacks to analyze video images. On the other hand, there is a limitation that the user must be proficient in the video detection system to collect reliable and accurate data. In addition, there is more financial liability for having a video detection system housed within the *Portable Video Detection System*. Nevertheless, as technology enhances the video detection systems, some of these limitations will not exist.

It is our anticipation that with the increasing computer power and improvements to visual processing algorithms, the current video detection systems installed directly in the data storage processing unit will be replaced by software alone. This would greatly

reduce the costs of the real-time operation of the *Portable Video Detection System*, especially if more than one system was used to collect data.

10.2 Detailed Specification and Guidelines

This section provides the detailed specifications for the individual components of the *Portable Video Detection System*. In addition, there are examples shown of each component to give a better understanding. Where these specifications describe portions of the system in general terms, but not in complete detail, only the best general practice is to prevail, and only materials and workmanship of the best quality should be used.

10.2.1 System Structure

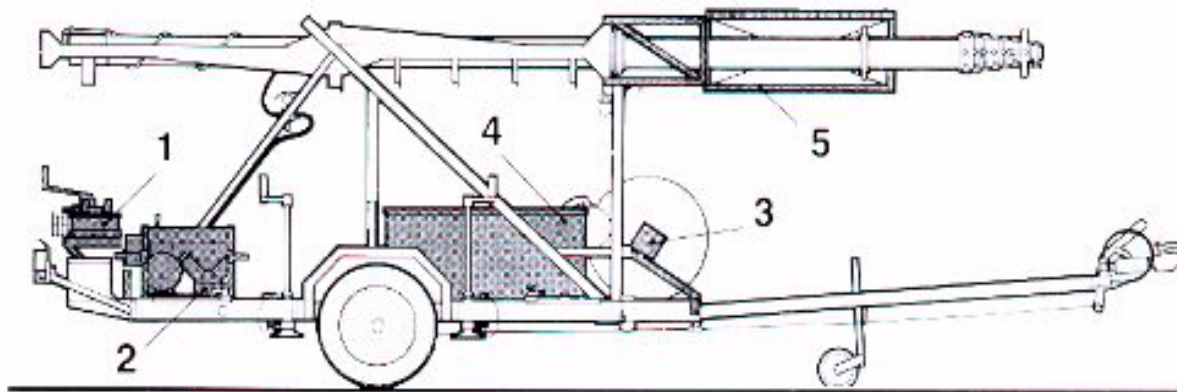
1. Trailer-mounted telescoping mast– The unit should be able to transport all exterior components of the Portable Video Detection System in a safe manner. In addition, it should be of suitable form and size to place at intersections. It should be provided with a minimum of the following items:
 - The trailer with mast should meet state guidelines for safe transport on Indiana roads;
 - The trailer should possess safety lights;
 - The trailer shall provide a standard female hitch for hauling;
 - The trailer and mast should be constructed of lightweight, strong materials, preferably high-strength aluminum;

- The trailer shall provide stabilizers to prevent the movement of the trailer in a parked position. The stabilizers should extend out from the trailer in a square or rectangular footprint pattern (Figure 10.7);
 - A mast shall be permanently attached onto the trailer while meeting state guidelines for safe transport on Indiana roads;
 - The mast should be made out of lightweight, strong materials, preferably high-strength aluminum;
 - The mast should be able to extend to a height of 50 ft.;
 - The mast being fully extended should be strong enough that it will not collapse during 50 mph winds;
 - The mast should be stable enough to allow only +/- 2 in. deflection at the top when all the necessary parts are attached (cameras, housings, mounts) at the maximum height of 50 ft. during 20 mph winds;
 - The peak of the mast should provide enough area (i.e. 0.25 ft²) to install the pan/tilt camera mounts;
 - Both the trailer and mast should not exceed a maximum weight of 3500 lbs;
 - Both the trailer and the mast should be water, rust, and corrosion proof, and painted an inconspicuous color (ex. Gray);
 - Both the trailer and mast should be able to withstand temperatures ranging from -30 °F to 120 °F;
2. Locking Weatherproof Electronics Housing – The housing protects components of the data acquisition system, data storage system, and power supply. It should be provided with a minimum of the following items:
- The housing should be constructed of lightweight, strong materials, preferably high-strength aluminum;
 - The housing should be constructed with rounded corners;
 - The housing should provide a safety and locking provisions (i.e. locks, cages) to prevent tampering with the mast or the housed equipment;

- The housing should have two separate compartments for the data acquisition/storage and power supply systems. In addition, easy access doors to should be provided for the compartments.
- The data acquisition compartment of the housing should provide shelves and cabinets for the electronic equipment;
- Both compartments should have separate ventilation systems with dust filters;
- The housing should be able to withstand temperatures ranging from -30 °F to 120 °F with an interior environmental control;
- The housing should be able to withstand 100% humidity conditions with an interior environmental control;
- The housing should be water, rust, and corrosion proof, and painted an inconspicuous color (ex. Gray);
- High Voltage Warning Indicator (Danger sign on housing).

Clark Masts (<http://www.alphalink.com.au/~gfs/Clark/clarkp11.htm>)

Clark Masts specializes in the design and manufacture of fast setting masts including stationary, vehicle mounted, and portable structures. They are located in Victoria, Australia, and their clients include both the public and private sector, and the military. They perform practically every manufacturing operation itself in-house including fabrication, casting, precision machining, plastic and rubber molding, anodizing and plating, painting and even the packing cases. The example below has an estimated cost of \$52,000.



OPTIONAL ITEMS

1. Winch Unit for Guys
(Standard on 802/30)
2. Compression Unit
220V AC or 12/24V
DC
3. Spare Wheel
4. Large Equipment Box
5. Operator Safety Cage
(Standard on 802/30)

Figure 10.6 Clark Model 802 Trailer-mounted Mast

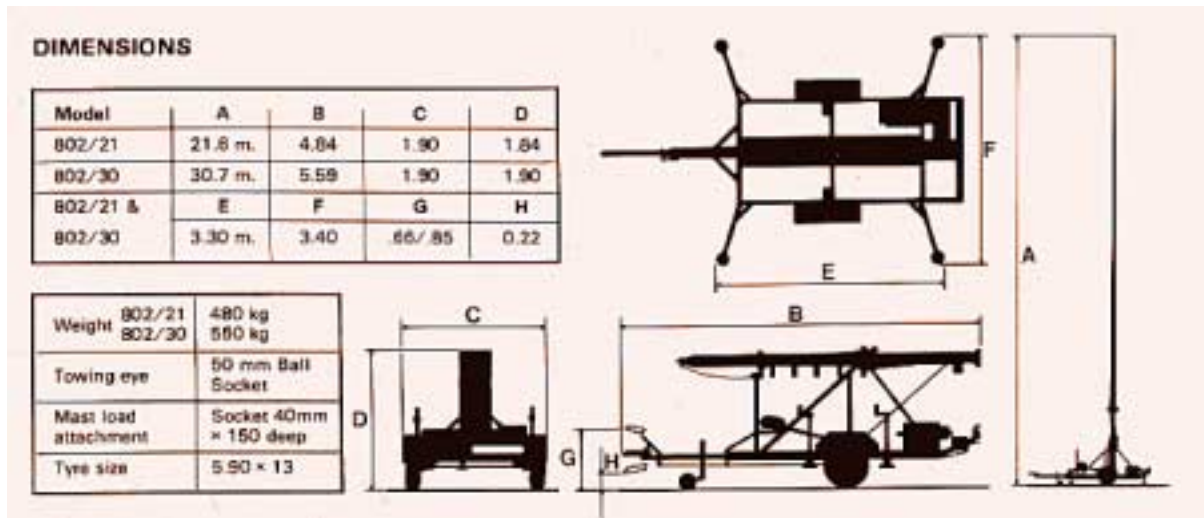


Figure 10.7 Model 802 Dimensions

SHIPPING

Both trailers may be knocked down for shipping.
Dimensions as follows :

Model 802/21

1 case 3050 x 1750 x 750 mm
weight gross 560 kg

1 case 4180 x 280 x 280 mm
weight gross 130 kg

Model 802/30

1 case 3050 x 1750 x 750 mm
weight gross 620 kg

1 case 5150 x 280 x 280 mm
weight gross 165 kg

Approximate re-assembly time for 2 men : 5 hours

BRIEF SPECIFICATION

Mast : Light alloy anodised, telescopic, air operated from petrol engine driven vee-twin compressor.

Trailer lighting : Conforming to current EEC Normes.

Mast extension and retraction controlled by single control lever. Mast turnable 360°.

Chassis : All tubular steel welded.

Brakes : Mechanical over-run and handparking.

Finish : Painted to Military Specification.

Standard colour : Yellow RAL1007.

PERFORMANCE

	802/21	802/30	
Headload, maximum recommended	40 kg	15 kg	Operating temperature limits : -30° (dry) to +45°C. Minimum operating crew : 1 person.
Time to set up and extend mast	5 min.	8 min.	
Max. windspeed unguyed with 0.5m ² side area load	35 km/hr	10 km/hr	
Max. windspeed guyed with 0.5m ² side area load	110 km/hr	80 km/hr	

Figure 10.8 Model 802 Specifications



Figure 10.9 Model 802 Example Setup

Floatograph Example (<http://www.floatograph.com/tower/>)

Floatograph specializes in the design and manufacture of surveillance masts including vehicle mounted and portable structures. They are located in Napa, California, and their clients include both the public and private sector, and the military. The example below has an estimated cost of \$26,000.



Figure 10.10 Floatograph Trailer-mounted mast

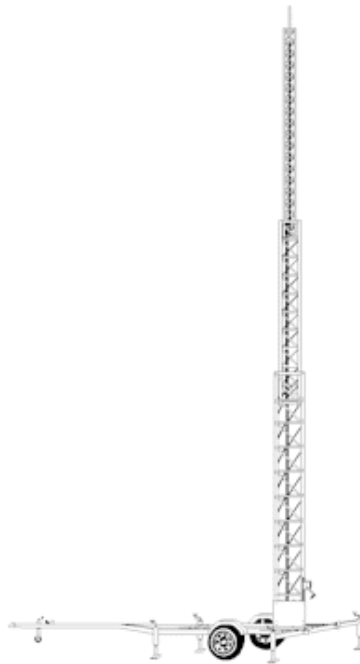


Figure 10.11 Floatograph Trailer-mounted mast setup

10.2.2 Data Acquisition System

Data Acquisition System-The data acquisition system should consist of the necessary parts to acquire video images for digital storage. This should include two digital video cameras, two environmental camera housings, a pan/tilt camera mount attachment, video and power cabling, and camera controller.

1. Digital Video Cameras – The digital video cameras obtain video images and shall be provided with a minimum of the following items:
 - The camera should have high-resolution of approximately 570 horizontal and 485 vertical lines. The resolution should not be lower than 350 horizontal and 265 vertical;
 - The camera may be a monochrome type but a switchable monochrome/color feature is recommended; a color camera is not recommended;
 - The camera should have auto-iris, motorized zoom lens, auto-focus, and a smaller focal length (i.e. 4mm) to achieve a wider image;
 - The camera should possess both the NTSC and PAL formats;
 - The camera should be able to provide remote control of camera functions including power, focus, and zoom control;
 - The camera should be able to modify a time and date display;
 - The camera should have a at least one BNC video output;
 - The camera should be able to attach to a pan-tilt camera mount;
 - The camera should be lightweight including all attachables (i.e. lenses, wires, and camera mount); recommended weight is less than 15 pounds;
 - The camera should be able to withstand cold and hot weather (-30 °F to 120 °F) and 100% humidity conditions; this condition applies to a camera protected by housing;
2. Camera Housing - The camera housing protects the camera from inclement weather, sun glare, and birds. It shall be provided with a minimum of the following items:

- The housing shall be compatible with both the video cameras and camera mount;
 - The housing should be lightweight and thermostatically controlled;
 - The housing should be able to provide remote control of the housing functions including power control and environmental control;
 - The housing should be able to withstand extreme cold and hot weather (-30 °F to 120 °F) and 100% humidity conditions with thermostatic control;
 - The housing should be water, rust, and corrosion proof.
3. Camera Mount - The camera mount provides controls the pan and tilt of the camera and shall be provided with a minimum of the following items:
- The camera mount shall be compatible with the cameras and camera housings;
 - The camera mount should provide 360° Pan and 180° Tilt capability at high speed;
 - The camera mount should be able to provide remote control of the mount functions including power, pan, and tilt;
 - The housing should be able to withstand extreme cold and hot weather (-30 °F to 120 °F) and 100% humidity conditions;
4. Video and Power Cabling - The cabling is used to remotely connect the video camera, housing, and mount to the camera controller, and it should be provided with a minimum of the following items:
- The cables should provide a large enough gauge to prevent easy wire fractures;
 - The cables should be made into a single integrated cable that contains all the power, video, and other cables from camera controller to the video camera, mount, and housing;
 - The cables should be able to withstand extreme cold and hot weather (-30 °F to 120 °F) and 100% humidity conditions;
 - The cables should be water, rust, and corrosion proof.
5. Camera Controller – The camera controller is used to control the camera, housing, and mount functions from a remote location and shall be provided with a minimum of the following items:

- The camera controller shall be compatible with the camera, housing, and mount;
- The controller should provide at least two video inputs, and two video outputs;
- The controller should be able to withstand extreme cold and hot weather (-30 °F to 120 °F) and 100% humidity conditions;

Panasonic Product Example (www.panasonic.com)

It is recommended that the data acquisition system be acquired from a single company for ease of component compatibility. Panasonic products provide such components, and include a camera, camera housing, camera controller, and pan/tilt zoom mechanism. The products shown in Figure 10.13 have a total estimated cost of \$5,900.

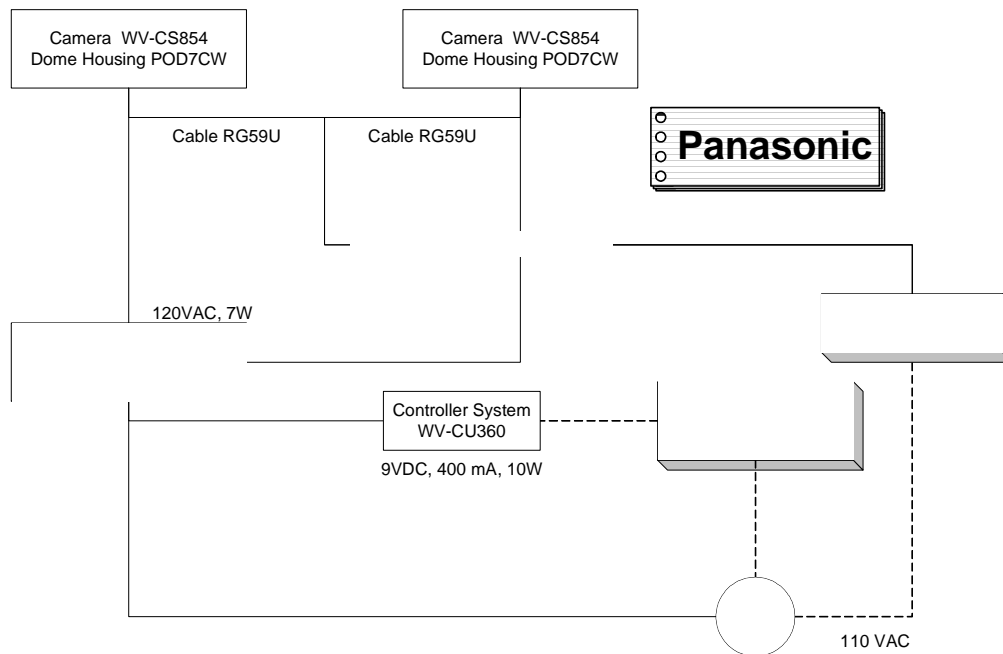


Figure 10.12 Panasonic data acquisition equipment



Integrated Camera with
Pan/Tilt and Housing
(WV-CS854)



Multiplexer (WJ-MP201)



Controller (WV-CU360)

Figure 10.13 Panasonic parts

10.2.3 Data Storage System

Data storage system – This is a specialized computer that is capable of storing continuous video images for sixteen hours and run software for video detection. It shall be provided with a minimum of the following items:

- The computer should contain enough hard disk, RAM, processing speed, and high-speed connection ports to collect sixteen hours of continuous digital video data. It is recommended that the ultra SCSI brands be used;
- The computer should possess a video graphics card that inputs and outputs both analog and digital video signal. In addition, the video graphics card should have compression no less than MJPEG;
- The computer should provide software, mouse, monitor, keyboard, floppy drive, zip drive, and CD-writer to control the video recording and video detection processes;
- The computer operating system should be able to read and write file sizes at least 1 Terabyte (i.e. Linux, Windows 2000, Windows NT);
- The computer should be housed in a case in the data acquisition compartment that allows for easy detachment and transport;

- The computer should be light enough for an average size person to carry for easy transport (Recommended weight less than 50 pounds);
- The computer should allow for network connection to other computers or archiving devices;
- The computer should be able to withstand extreme cold and hot weather (-30 °F to 120 °F) and 100% humidity conditions.

Computer Hardware and Software Examples

Below is an example list of computer products needed to build a data storage system. The important components include an extensive amount of hard disk space, a digital video card that can input and output both digital and analog video data, and an operating system that is able to create large file sizes. Altogether, it has an estimated cost of \$5,000

- AMD Athlon “Thunderbird” 1.2 GHz;
- 1536 MB PC 133 CAS2 RAM;
- Cheetah 73.4GB Ultra 160 SCSI hard disk (2);
- DC 500 plus video capture card (Pinnacle Systems);
- Windows 2000;
- Computer Chassis;
- 1.44MB Floppy Drive;
- Adaptec 29160 SCSI Card – 160MB
- Etherling 10-100 PCI;
- CD Rewritable;
- Etherfast 8-Port 10-100;

10.2.4 Power Supply

Power Supply – The power supply should give electricity to all necessary equipment for up to 48 hours and it shall be provided with a minimum of the following items:

- The power supply should provide enough electricity to power all the equipment for 48 hours;
- The power supply compartment of the system structure weather proof housing should be able to safely protect and cover the power supply;
- The power supply should be able to withstand extreme cold and hot weather (-30 °F to 120 °F) and 100% humidity conditions.

Power Generator Example (<http://www.hondapowerequipment.com/gen.htm>)

An option for the power supply is a quiet, economical, and efficient generator that is able to supply power to sensitive electronics and computer equipment. Honda, Inc. provides such a product, the Honda3000is, which has an estimated cost of \$2,000



Figure 10.14 Honda 3000is Power Generator

Table 10.1 Honda 3000is Specifications

Specifications	EU3000is
Engine	6.5 HP single cylinder, overhead valve, air cooled
Displacement	196 cc
AC output	120V; 3000W max (25A); 2800W rated (23.3A);
DC output	12V, 144W (12A)
Starting System	Recoil, Electric

Table 10.1 Continued

Fuel-tank capacity	3.4 gallons
Run time tankful	7.2 hrs. @ rated load 20 hrs. @ 1/4 load
Dimensions (L x W x H)	25.8" x 18.9" x 22.4":
Noise Level	58 dB @ rated load 49 dB @ 1/4 load
Dry Weight	134 lbs
Estimated Cost	\$2000

Deep Cycle Battery Example (<http://www.ee.ualberta.ca/~schmaus/dcbat.html#bother>)

The two most common types of deep cycle batteries are flooded cell and valve regulated. Flooded cell deep cycle batteries are divided into low maintenance (the most common) and maintenance free. The advantages of maintenance free batteries are less preventative maintenance, small water loss, faster recharging, greater overcharge resistance, reduced terminal corrosion, 1.5 times more life cycles, and up to 200% less self discharge. However, they are more prone to deep discharge (dead battery) failures

and if sealed, a shorter life in hot climates because lost water cannot be replaced.

Maintenance free batteries are generally more expensive than low maintenance batteries.

Valve Regulated Lead Acid (VRLA) batteries are generally divided into two groups, gel cell and Absorbed Glass Mat (AGM). VRLA batteries are spill proof, totally maintenance free, and have a longer shelf life. Their disadvantage is the high initial cost (two to three times), but they could have an overall lower cost due to a longer lifetime and no "watering" labor costs, if properly maintained and recharged.

The price for deep-cycle batteries can range up to a \$1000.00 or more, depending on the type. The typical flooded cell deep cycle batteries are around \$100.00, and the typical VRLA batteries are approximately \$400.00. In addition, one should consider the extra load a set of deep cycle batteries can have on the trailer of the system structure of the *Portable Video Detection System*, and should be included in the design of the trailer unit.

10.2.5 Video Detection System

Video Detection System – The video detection system should be the Autoscope 2004 system as recommended, but not required. It shall be provided with a minimum of the following items:

- The video detection system should be compatible with the data storage system, and be obtained from local vendor;
- The video detection system should include all hardware, software, wires, installation manuals and operating manuals;

Video Detection System Examples

The video detection systems have a price ranging from \$30,000 - \$50,000.



Figure 10.15 Vantage Plus by Odectics
<http://www.midamsignal.com/videotrak/video.htm>



Figure 10.16 Autoscope 2004LE by Econolite Control Products, Inc.
<http://www.econolite.com/product/autoscope/2004LE.htm>



Figure 10.17 Autoscope 2004LE by Econolite Control Products, Inc.
<http://www.midamsignal.com/videotrak/video.htm>

11. CLOSURE

This research project investigated the potential capabilities of two selected video detection systems for counting and classifying turning vehicles at intersections. The detection systems selected Autoscope 2004 and VideoTrak 905. The mobile traffic laboratory and mounted pan/tilt/zoom camera were used to collect 76 hours of videotape data at six intersections in the Lafayette Area, Indiana. Subsequently, the flow conservation and tracking strip methods have been developed for the Autoscope and VideoTrak systems, respectively, which utilize output generated by these systems. Ultimately, after a comprehensive comparative evaluation of the two systems Autoscope is recommended for near-future application. Finally, general specifications were developed to help build a prototype system.

The Autoscope 2004 system was selected for this study because it provided satisfactory spot detection capabilities and is among technologies endorsed by INDOT. Our research confirmed that the Autoscope system was a flexible tool that can be used a wide variety of locations including intersections. The VideoTrak system was selected for evaluation because of its tracking capabilities. Our study has indicated that the tracking can only be done in either horizontally or vertically, which limits the VideoTrak usefulness for counting turning vehicles. Except for the tracking strip method, VideoTrak standard features must have signal status input for counting turning volumes at intersections.

The flow conservation method of extracting turning flows from multiple detector counts as proposed is valid. The limitation of the technique is where the number of spots with sufficiently diversified set of turning flows is too small to extract all the turning flows. Intersections with single-lane approaches are an example. On the other hand, intersections with exclusive lanes for either left or right-turning flows can enhance the proposed estimation. In addition, it can be combined with any detection technique that allows fast setting of multiple detectors with localized detection spots.

Results indicate that using Autoscope together with the flow conservation method is feasible. It should be kept in mind though, that it is challenging to locate spot detectors that are passed by all vehicles of a certain flow. In addition, the quality of detector counts is critical for the quality of the turning flow estimates. Our findings show that one spot detector that gives false detections can greatly influence the overall turning volume estimations.

The tracking strip method developed exclusively for the Academia version of the VideoTrak system feasible. In fact, this technique outperformed the Autoscope system for accurately estimating turning movement counts at intersections. However, the main drawback of this technology lies in the limitation on the number of tracking strips, which does not allow for efficient extraction of data from the image. VideoTrak has the potential to be very successful in estimating turning movements if this limitation is removed.

This research concluded with specifications for a portable video detection system that can store and analyze up to sixteen hours of videotape data. They describe a portable trailer unit with mounted mast and equipment that can operate unattended and

require no more than one person to setup. It is designed as a modular system that allows the integration of other components and functions for traffic detection, surveillance, and monitoring.

The specifications for the portable video detection system present two possible modes of operation. Foremost, the system can be used with current video detection technology for off-line analysis of video images. For instance, the video images are digitally stored onto a specialized computer where they then can be transmitted to an off-line video detection system for analysis. Secondly, it could also be utilized to perform real-time data analysis. However, current video detection technology does not economically allow real-time analysis using the portable video detection. It is thought that with increasing technology this type of operation can successfully be done within several years.

LIST OF REFERENCES

LIST OF REFERENCES

- Autoscope Std (2001). Econolite Control Products, Inc.
<http://www.econolite.com/product/autoscope/2004LE.htm>
- Bahler, S., Kranig, J., Minge, E., (1998). Field Test of Nonintrusive Traffic Detection Technologies. Transportation Research Record, Issue 1643, pp. 161-170.
- Coifman, B, Beymer, D, McLauchlan, P, Malik, J, (1998). A Real-Time Computer Vision System for Vehicle Tracking and Traffic Surveillance. Transportation Research Part C 6, pp. 271-288.
- Cottrell, B.H., (1994). Evaluation of a Video Image Detection System (VIDS) – Final Report. <http://ntl.bts.gov/DOCS/cottre.html>.
- Cremer, M. and H. Keller (1987). A New Class of Dynamic Methods for the Identification of Origin-destination Flows, Transportation Research, Part B, 21B(2), pp. 117-132.
- Hauer, E., E. Pagitsas, and B. T. Shin (1981). Estimation of Turning Flows from Automatic Counts, Transportation Research Record 795, Transportation Research Board, Washington, D.C., pp. 1-7.
- Lu, Y.-J., Y.-H. Hsu, and G. C. Tan (1988). Application of the Image Analysis Technique to Detect Left-Turning Vehicles at Intersections, Transportation Research Record 1194, Transportation Research Board, Washington, D.C., pp. 120-128.
- Middleton, D., Parker, R., (2000). Initial Evaluation of Selected Detectors to Replace Inductive Loops on Freeways. Texas Transportation Institute, Research Report 1439-7.
- Mountain, L.J. and P. Westwell (1983). The Accuracy of Estimation of Turning Flows from Automatic Counts, Traffic Engineering and Control, No. 1, Vol. 24, pp. 3-7.
- Ploss, G. and H. Keller (1986). Dynamic Estimation of Origin and Destination Flows from Traffic Counts in Networks, Proceedings of International Conference on Transportation Systems Studies, Tata McGraw-Hill, Delhi, India, pp. 211-221.
- SAS Institute (2001). Statistical Analysis Software Technical Support.
<http://www.sas.com/service/techsup/intro.html>

Van Zuylen, H. J., (1979). The Estimation of Turning Flows on a Junction, Traffic Engineering and Control, No. 20, Vol. 11, pp. 539-541.

VideoTrak 905 (2001). Peek Traffic.

<http://www.peek-traffic.com/ptsi/products/videotrak-905/datasheet.html>

Vikler, M.R., and N. Kumar (1998). System to identify turning movements at signalized intersections. Journal of Transportation Engineering 124: pp. 607-609

APPENDIX

'This program is used to estimate turning volumes from Academia output using Visual Basic language.
 'It connects to a Microsoft Access Database where the output is stored to retrieve information.
 'First a user must enter necessary Strip information for the Academia out.
 'This includes: the number of strips to be analyzed, if the strips are horizontal or vertical,
 'if the strips are tracking right/down or left/right, the pixel range that define the entrance
 'to the strip, the range to define right turn movement, the pixel range to define left turn movement,
 'the pixel range to define through movement, and the pixel range used to classify vehicles.
 'Once those information are stored in the gblfile, gbltable, gblStrip, and gblStrips variables,
 'the function analysis will estimate turning volumes and store them in an array called tapID.

Option Explicit

Public gblfile As String 'Variable used to declare Microsoft Database *.mdb filename
 Public gbltable As String 'Variable used to declare table within Microsoft Database *.mdb file
 Public gblStrip As Integer 'Variable used to declare total number of strips

Public gblStrips(4, 11) As Integer 'Array used to declare strip attributes; (0-4) represents Strip ID number in Academia output

'(0-11) represent input variables HV, XX, E1, E2, R1, R2, L1, L2, Th1, Th2, CL1, CL2,
 Public HV As Integer 'Variable used to in gblStips(4, 11) array declare Horizontal(0)or Vertical Strip(1).
 Index = 0.

Public XX As Integer 'Variable used to in gblStips(4, 11) array declare strip track left/up(0) or
 down/right(1) Index = 1.

Public E1 As Integer 'Variable used to in gblStips(4, 11) array declare pixel group boundary for vehicles
 entering strip. Must be entered by user. Index = 2.

Public E2 As Integer 'Variable used to in gblStips(4, 11) array declare pixel group boundary for vehicles
 entering strip. Must be entered by user. Index = 3.

Public R1 As Integer 'Variable used to in gblStips(4, 11) array declare pixel group boundary for vehicles
 exiting strip for Right Turn. Must be entered by user. Index = 4.

Public R2 As Integer 'Variable used to in gblStips(4, 11) array declare pixel group boundary for vehicles
 exiting strip for Right Turn. Must be entered by user. Index = 5.

Public L1 As Integer 'Variable used to in gblStips(4, 11) array declare pixel group boundary for vehicles
 exiting strip for Left Turn. Must be entered by user. Index = 6.

Public L2 As Integer 'Variable used to in gblStips(4, 11) array declare pixel group boundary for vehicles
 exiting strip for Left Turn. Must be entered by user. Index = 7.

Public Th1 As Integer 'Variable used to in gblStips(4, 11) array declare pixel group boundary for vehicles
 exiting strip for Through Turn. Must be entered by user. Index = 8.

Public Th2 As Integer 'Variable used to in gblStips(4, 11) array declare pixel group boundary for vehicles
 exiting strip for Through Turn. Must be entered by user. Index = 9.

Public CL1 As Integer 'Variable used to in gblStips(4, 11) array declare pixel group boundary to classify
 vehicles within strip. Must be entered by user. Index = 10.

Public CL2 As Integer 'Variable used to in gblStips(4, 11) array declare pixel group boundary to classify
 vehicles within strip. Must be entered by user. Index = 11.

Public tapID(8, 4, 11) As Integer 'Array used to output turning movements to a file. (0-8) represents Tap
 number given from Acamdemia output. Index = 12.

'The Tap number is used to interval data during collection. Only 10 intervals are possible, so the max value
 is 9.

'(0-4) represents Strip ID number in Academia output. (0-11) is used to hold varibles to count and classify
 vehicles.

Public lcar As Integer 'Variable index used in tap ID(8, 4, 11) array set to 0.

Public rcar As Integer 'Variable index used in tap ID(8, 4, 11) array set to 1.

Public tcar As Integer 'Variable index used in tap ID(8, 4, 11) array set to 2.

Public ltruck As Integer 'Variable index used in tap ID(8, 4, 11) array set to 3.

Public rtruck As Integer 'Variable index used in tap ID(8, 4, 11) array set to 4.

Public ttruck As Integer 'Variable index used in tap ID(8, 4, 11) array set to 5.

Public lsemi As Integer 'Variable index used in tap ID(8, 4, 11) array set to 6.
Public rsemi As Integer 'Variable index used in tap ID(8, 4, 11) array set to 7.
Public tsemi As Integer 'Variable index used in tap ID(8, 4, 11) array set to 8.
Public lttotal As Integer 'Variable index used in tap ID(8, 4, 11) array set to 9.
Public rttotal As Integer 'Variable index used in tap ID(8, 4, 11) array set to 10.
Public tttotal As Integer 'Variable index used in tap ID(8, 4, 11) array set to 11.
'The beginning of this program should have some input method for the gblfile
'gbltable, gblStrip, and gblStrips variables
'Function that extracts turning volumes from Academia output.
Sub analysis()

Dim db As ADODB.Connection 'Variable db used as Microsoft Database connection
Dim strConn As String 'Variable to set type of database connection
Dim rsData As ADODB.Recordset 'Variable used to define recordset for database
'Dim strsql As String--not used
'Dim strTable As String--not used

Dim A(31, 10) As Integer 'Temporary Array that holds the previous individual record information for individual VehID's
'from the Academia Output. (0-31) represent VehID number in Academia output. (0-10) are index variables
'the information from the Academia output. They are tap(0), vehid(1), strip(2), track(3), sx(4),
'sy(5), ex(6), ey(7), length(8), speed(9), count(10)
Dim tap As Integer 'Variable index used in A(31,10) array set to 0.
Dim vehid As Integer 'Variable index used in A(31,10) array set to 0.
Dim strip As Integer 'Variable index used in A(31,10) array set to 0.
Dim track As Integer 'Variable index used in A(31,10) array set to 0.
Dim sx As Integer 'Variable index used in A(31,10) array set to 0.
Dim sy As Integer 'Variable index used in A(31,10) array set to 0.
Dim ex As Integer 'Variable index used in A(31,10) array set to 0.
Dim ey As Integer 'Variable index used in A(31,10) array set to 0.
Dim length As Integer 'Variable index used in A(31,10) array set to 0.
Dim speed As Integer 'Variable index used in A(31,10) array set to 0.
Dim count As Integer 'Variable index used in A(31,10) array set to 0.

Dim d_array(9) As Integer 'Temporary Array that holds the current individual record information for individual VehID's
'from the Academia Output. (0-9) are index variables the information from the Academia output. They are
tap(0), vehid(1),
'strip(2), track(3), sx(4), sy(5), ex(6), ey(7), length(8), speed(9),

Dim i As Integer 'Variable used for counters in loops
Dim j As Integer 'Variable used for counters in loops
Dim k As Integer 'Variable used for counters in loops

'Initialize index values within gblStrips Arrays
HV = 0
E1 = 1
E2 = 2
R1 = 3
R2 = 4
L1 = 5
L2 = 6
Th1 = 7
Th2 = 8
XX = 9
CL1 = 10

CL2 = 11

'Initialize index values within tapID Arrays

```
lcar = 0
ltruck = 1
lsemi = 2
ltotal = 3
tcar = 4
ttruck = 5
tsemi = 6
ttotal = 7
rcar = 8
rtruck = 9
rsemi = 10
rtotal = 11
```

'Initialize index values within the d_array and A Arrays

```
tap = 0
vehid = 1
strip = 2
track = 3
sx = 4
sy = 5
ex = 6
ey = 7
length = 8
speed = 9
count = 10
```

' Initialize A array values with a possible (0-31) vehicles as outputted by Academia

```
For i = 0 To 31
  A(i, tap) = 0
  A(i, vehid) = i
  A(i, strip) = 0
  A(i, track) = 0
  A(i, sx) = 0
  A(i, sy) = 0
  A(i, ex) = 0
  A(i, ey) = 0
  A(i, length) = 0
  A(i, speed) = 0
  A(i, count) = 0
Next i
```

'Initialize tapID Array values to zero

```
For i = 0 To 8
  For j = 0 To 4
    For k = 0 To 11
      tapID(i, j, k) = 0
    Next k
  Next j
Next i
```

Next i

```
'Specify the type of database connection  
strCnn = "Provider=Microsoft.Jet.OLEDB.4.0; Data Source=" & gblfile
```

```
'Open the database connection  
Set db = New ADODB.Connection  
db.Open strCnn
```

```
'Open the table within a database and read first recordset  
Set rsData = New ADODB.Recordset  
rsData.Open gbltable, db, adOpenForwardOnly, adLockReadOnly
```

```
'Read each recordset in the database sequentially  
With rsData
```

```
    If Not rsData.EOF Then rsData.MoveNext 'Ask if beginning of file.
```

```
    Do Until rsData.EOF 'Do until end of file.
```

```
        'Read first record data fields from database and store in d_array  
        d_array(tap) = rsData.Fields(1)  
        d_array(vehid) = rsData.Fields(2)  
        d_array(strip) = rsData.Fields(3)  
        d_array(track) = rsData.Fields(4)  
        d_array(sx) = rsData.Fields(5)  
        d_array(sy) = rsData.Fields(6)  
        d_array(ex) = rsData.Fields(7)  
        d_array(ey) = rsData.Fields(8)  
        d_array(length) = rsData.Fields(10)  
        d_array(speed) = rsData.Fields(11)
```

```
        ' Check to see if it is the first time this VehID is being tracked  
        If A(d_array(vehid), track) = 0 Then
```

```
            ' Is the strip tracking up or left?  
            If gblStrips(d_array(strip), XX) = 0 Then
```

```
                'Put current record data into A array  
                A(d_array(vehid), tap) = d_array(tap)  
                A(d_array(vehid), vehid) = d_array(vehid)  
                A(d_array(vehid), strip) = d_array(strip)  
                A(d_array(vehid), track) = d_array(track)  
                A(d_array(vehid), sx) = d_array(sx)  
                A(d_array(vehid), sy) = d_array(sy)  
                A(d_array(vehid), ex) = d_array(ex)  
                A(d_array(vehid), ey) = d_array(ey)  
                A(d_array(vehid), length) = d_array(length)  
                A(d_array(vehid), speed) = d_array(speed)  
                A(d_array(vehid), count) = 1
```

```
            ' Is the strip tracking down or right?  
            ElseIf gblStrips(d_array(strip), XX) = 1 Then
```

```

'Put current record data into A array
A(d_array(vehid), tap) = d_array(tap)
A(d_array(vehid), vehid) = d_array(vehid)
A(d_array(vehid), strip) = d_array(strip)
A(d_array(vehid), track) = d_array(track)
A(d_array(vehid), sx) = d_array(sx)
A(d_array(vehid), sy) = d_array(sy)
A(d_array(vehid), ex) = d_array(sx)
A(d_array(vehid), ey) = d_array(sy)
A(d_array(vehid), length) = d_array(length)
A(d_array(vehid), speed) = d_array(speed)
A(d_array(vehid), count) = 1

End If

' Is it a vehicle that is being tracked as represented by vehid
ElseIf A(d_array(vehid), track) < d_array(track) And A(d_array(vehid), strip) = d_array(strip) Then

' Is the record from a horizontal tracking strip?
If gblStrips(A(d_array(vehid), strip), HV) = 0 Then

' Is the strip tracking up or left?
If gblStrips(d_array(strip), XX) = 0 Then

'Exchange previous track, sx, sy information of A array with d array
A(d_array(vehid), track) = d_array(track)
A(d_array(vehid), sx) = d_array(sx)
A(d_array(vehid), sy) = d_array(sy)

' Does the starting edge pixel lie within the classifying pixel group of that strip
If (gblStrips(A(d_array(vehid), strip), CL1) < A(d_array(vehid), sy)) And
(gblStrips(A(d_array(vehid), strip), CL2) > A(d_array(vehid), sy)) Or _
(gblStrips(A(d_array(vehid), strip), CL1) > A(d_array(vehid), sy)) And
(gblStrips(A(d_array(vehid), strip), CL2) < A(d_array(vehid), sy)) Then

'Begin summing up lengths to be averaged later on.
A(d_array(vehid), length) = A(d_array(vehid), length) + d_array(length)
A(d_array(vehid), count) = A(d_array(vehid), count) + 1

End If

' Is the strip tracking down or right?
ElseIf gblStrips(d_array(strip), XX) = 1 Then

'Exchange previous track, ex, ey information of A array with d array
A(d_array(vehid), track) = d_array(track)
A(d_array(vehid), sx) = d_array(ex)
A(d_array(vehid), sy) = d_array(ey)

' Does the starting edge pixel lie within the classifying pixel group of that strip
If (gblStrips(A(d_array(vehid), strip), CL1) < A(d_array(vehid), sy)) And
(gblStrips(A(d_array(vehid), strip), CL2) > A(d_array(vehid), sy)) Or _
(gblStrips(A(d_array(vehid), strip), CL1) > A(d_array(vehid), sy)) And
(gblStrips(A(d_array(vehid), strip), CL2) < A(d_array(vehid), sy)) Then

```

```

    'Begin summing up lengths to be averaged later on.
    A(d_array(vehid), length) = A(d_array(vehid), length) + d_array(length)
    A(d_array(vehid), count) = A(d_array(vehid), count) + 1

End If

End If

' Is the record from a vertical tracking strip?
ElseIf gblStrips(A(d_array(vehid), strip), HV) = 1 Then

    ' Is the strip tracking up or left?
    If gblStrips(d_array(strip), XX) = 0 Then

        'Exchange previous track, sx, sy information of A array with d array
        A(d_array(vehid), track) = d_array(track)
        A(d_array(vehid), sx) = d_array(sx)
        A(d_array(vehid), sy) = d_array(sy)

        'Does the ending edge pixel lie within the classifying pixel group of that strip
        If (gblStrips(A(d_array(vehid), strip), CL1) < A(d_array(vehid), sx)) And
(gblStrips(A(d_array(vehid), strip), CL2) > A(d_array(vehid), sx)) Or _
(gblStrips(A(d_array(vehid), strip), CL1) > A(d_array(vehid), sx)) And
(gblStrips(A(d_array(vehid), strip), CL2) < A(d_array(vehid), sx)) Then

            'Begin summing up lengths to be averaged later on.
            A(d_array(vehid), length) = A(d_array(vehid), length) + d_array(length)
            A(d_array(vehid), count) = A(d_array(vehid), count) + 1

        End If

    ' Is the strip tracking down or right?
    ElseIf gblStrips(d_array(strip), XX) = 1 Then

        'Exchange previous track, sx, sy information of A array with d array
        A(d_array(vehid), track) = d_array(track)
        A(d_array(vehid), sx) = d_array(ex)
        A(d_array(vehid), sy) = d_array(ey)

        'Does the starting edge pixel lie within the classifying pixel group of that strip
        If (gblStrips(A(d_array(vehid), strip), CL1) < A(d_array(vehid), sx)) And
(gblStrips(A(d_array(vehid), strip), CL2) > A(d_array(vehid), sx)) Or _
(gblStrips(A(d_array(vehid), strip), CL1) > A(d_array(vehid), sx)) And
(gblStrips(A(d_array(vehid), strip), CL2) < A(d_array(vehid), sx)) Then

            'Begin summing up lengths to be averaged later on.
            A(d_array(vehid), length) = A(d_array(vehid), length) + d_array(length)
            A(d_array(vehid), count) = A(d_array(vehid), count) + 1

        End If

    End If

End If

```

```

End If

' Is there a same vehid within a differ strip that might be mistaken the current vehID being tracked?
ElseIf A(d_array(vehid), track) < d_array(track) And A(d_array(vehid), strip) <> d_array(strip) Then

    ' Is the record from a horizontal tracking strip?
    If gblStrips(A(d_array(vehid), strip), HV) = 0 Then

        'Does the ending edge pixel lie within the classifying pixel group of that strip
        If (gblStrips(A(d_array(vehid), strip), E1) < A(d_array(vehid), ey)) And
(gblStrips(A(d_array(vehid), strip), E2) > A(d_array(vehid), ey)) Or _
(gblStrips(A(d_array(vehid), strip), E1) > A(d_array(vehid), ey)) And
(gblStrips(A(d_array(vehid), strip), E2) < A(d_array(vehid), ey)) Then

            'Does the ending edge pixel lie within the left classifying pixel group of that strip
            If (gblStrips(A(d_array(vehid), strip), L1) < A(d_array(vehid), sy)) And
(gblStrips(A(d_array(vehid), strip), L2) > A(d_array(vehid), sy)) Or _
(gblStrips(A(d_array(vehid), strip), L1) > A(d_array(vehid), sy)) And
(gblStrips(A(d_array(vehid), strip), L2) < A(d_array(vehid), sy)) Then

                'Average the vehicle summed up vehicle lengths of the A array!
                A(d_array(vehid), length) = (A(d_array(vehid), length)) / (A(d_array(vehid), count))

                'Classify the vehicle according to the first length classification, you can change the
numbers!
                If A(d_array(vehid), length) > 0 And A(d_array(vehid), length) <= 28 Then

                    'Increase movement counter of tapID
                    tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), lcar) = tapID(A(d_array(vehid),
tap), A(d_array(vehid), strip), lcar) + 1

                'Classify the vehicle according to second pre-specified length classification, you can
change the numbers!
                ElseIf A(d_array(vehid), length) > 28 And A(d_array(vehid), length) <= 33 Then

                    'Increase movement counter of tapID
                    tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), ltruck) =
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), ltruck) + 1

                'Classify the vehicle according to third pre-specified length classification, you can change
the numbers!
                ElseIf A(d_array(vehid), length) > 33 Then

                    'Increase movement counter of tapID
                    tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), lsemi) =
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), lsemi) + 1

            End If

            'Create subtotal for the left movement
            tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), ltotal) = tapID(A(d_array(vehid),
tap), A(d_array(vehid), strip), lcar) + tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), ltruck) +
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), lsemi)

```

```

'Does the ending edge pixel lie within the right classifying pixel group of that strip
ElseIf (gblStrips(A(d_array(vehid), strip), R1) < A(d_array(vehid), sy)) And
(gblStrips(A(d_array(vehid), strip), R2) > A(d_array(vehid), sy)) Or _
(gblStrips(A(d_array(vehid), strip), R1) > A(d_array(vehid), sy)) And
(gblStrips(A(d_array(vehid), strip), R2) < A(d_array(vehid), sy)) Then

'Average the vehicle summed up vehicle lengths of the A array!
A(d_array(vehid), length) = (A(d_array(vehid), length)) / (A(d_array(vehid), count))

'Classify the vehicle according to the first length classification, you can change the
numbers!
If A(d_array(vehid), length) > 0 And A(d_array(vehid), length) <= 28 Then

'Increase movement counter of tapID
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), rcar) = tapID(A(d_array(vehid),
tap), A(d_array(vehid), strip), rcar) + 1

'Classify the vehicle according to the second length classification, you can change the
numbers!
ElseIf A(d_array(vehid), length) > 28 And A(d_array(vehid), length) <= 33 Then

'Increase movement counter of tapID
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), rtruck) =
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), rtruck) + 1

'Classify the vehicle according to the third length classification, you can change the
numbers!
ElseIf A(d_array(vehid), length) > 33 Then

'Increase movement counter of tapID
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), rsemi) =
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), rsemi) + 1

End If

'Create subtotal for the right movement
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), rtotal) = tapID(A(d_array(vehid),
tap), A(d_array(vehid), strip), rcar) + tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), rtruck) +
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), rsemi)

'Does the ending edge pixel lie within the through classifying pixel group of that strip
ElseIf (gblStrips(A(d_array(vehid), strip), Th1) < A(d_array(vehid), sy)) And
(gblStrips(A(d_array(vehid), strip), Th2) > A(d_array(vehid), sy)) Or _
(gblStrips(A(d_array(vehid), strip), Th1) > A(d_array(vehid), sy)) And
(gblStrips(A(d_array(vehid), strip), Th2) < A(d_array(vehid), sy)) Then
'Does the last start pixel lie within the thru turn area of that strip

'Average the vehicle summed up vehicle lengths of the A array!
A(d_array(vehid), length) = (A(d_array(vehid), length)) / (A(d_array(vehid), count))

'Classify the vehicle according to the first length classification, you can change the
numbers!
If A(d_array(vehid), length) > 0 And A(d_array(vehid), length) <= 28 Then

```

```

        'Increase movement counter of tapID
        tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), tcar) = tapID(A(d_array(vehid),
tap), A(d_array(vehid), strip), tcar) + 1

```

```

        'Classify the vehicle according to the second length classification, you can change the
numbers!

```

```

        ElseIf A(d_array(vehid), length) > 28 And A(d_array(vehid), length) <= 33 Then

```

```

            'Increase movement counter of tapID
            tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), ttruck) =
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), ttruck) + 1

```

```

            'Classify the vehicle according to the third length classification, you can change the
numbers!

```

```

            ElseIf A(d_array(vehid), length) > 33 Then

```

```

                'Increase movement counter of tapID
                tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), tsemi) =
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), tsemi) + 1

```

```

            End If

```

```

        'Create subtotal for the through movement
        tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), tttotal) = tapID(A(d_array(vehid),
tap), A(d_array(vehid), strip), tcar) + tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), ttruck) +
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), tsemi)

```

```

    End If

```

```

End If

```

```

' Is the record from a vertical tracking strip?
ElseIf gblStrips(A(d_array(vehid), strip), HV) = 1 Then

```

```

    'Does the ending edge pixel lie within the classifying pixel group of that strip
    If (gblStrips(A(d_array(vehid), strip), E1) < A(d_array(vehid), ex)) And
(gblStrips(A(d_array(vehid), strip), E2) > A(d_array(vehid), ex)) Or _
(gblStrips(A(d_array(vehid), strip), E1) > A(d_array(vehid), ex)) And
(gblStrips(A(d_array(vehid), strip), E2) < A(d_array(vehid), ex)) Then

```

```

        'Does the ending edge pixel lie within the left classifying pixel group of that strip
        If (gblStrips(A(d_array(vehid), strip), L1) < A(d_array(vehid), sx)) And
(gblStrips(A(d_array(vehid), strip), L2) > A(d_array(vehid), sx)) Or _
(gblStrips(A(d_array(vehid), strip), L1) > A(d_array(vehid), sx)) And
(gblStrips(A(d_array(vehid), strip), L2) < A(d_array(vehid), sx)) Then

```

```

            'Average the vehicle summed up vehicle lengths of the A array!
            A(d_array(vehid), length) = (A(d_array(vehid), length)) / (A(d_array(vehid), count))

```

```

            'Classify the vehicle according to the first length classification, you can change the
numbers!

```

```

            If A(d_array(vehid), length) > 0 And A(d_array(vehid), length) <= 28 Then

```

```

                'Increase movement counter of tapID

```

```
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), lcar) = tapID(A(d_array(vehid),
tap), A(d_array(vehid), strip), lcar) + 1
```

```
'Classify the vehicle according to the second length classification, you can change the
numbers!
```

```
ElseIf A(d_array(vehid), length) > 28 And A(d_array(vehid), length) <= 33 Then
```

```
'Increase movement counter of tapID
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), ltruck) =
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), ltruck) + 1
```

```
'Classify the vehicle according to the third length classification, you can change the
numbers!
```

```
ElseIf A(d_array(vehid), length) > 33 Then
```

```
'Increase movement counter of tapID
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), lsemi) =
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), lsemi) + 1
```

```
End If
```

```
'Create subtotal for the left movement
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), ltotal) = tapID(A(d_array(vehid),
tap), A(d_array(vehid), strip), lcar) + tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), ltruck) +
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), lsemi)
```

```
'Does the ending edge pixel lie within the right classifying pixel group of that strip
ElseIf (gblStrips(A(d_array(vehid), strip), R1) < A(d_array(vehid), sx)) And
(gblStrips(A(d_array(vehid), strip), R2) > A(d_array(vehid), sx)) Or _
(gblStrips(A(d_array(vehid), strip), R1) > A(d_array(vehid), sx)) And
(gblStrips(A(d_array(vehid), strip), R2) < A(d_array(vehid), sx)) Then
```

```
'Average the vehicle summed up vehicle lengths of the A array!
A(d_array(vehid), length) = (A(d_array(vehid), length)) / (A(d_array(vehid), count))
```

```
'Classify the vehicle according to the first length classification, you can change the
numbers!
```

```
If A(d_array(vehid), length) > 0 And A(d_array(vehid), length) <= 28 Then
```

```
'Increase movement counter of tapID
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), rcar) = tapID(A(d_array(vehid),
tap), A(d_array(vehid), strip), rcar) + 1
```

```
'Classify the vehicle according to the second length classification, you can change the
numbers!
```

```
ElseIf A(d_array(vehid), length) > 28 And A(d_array(vehid), length) <= 33 Then
```

```
'Increase movement counter of tapID
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), rtruck) =
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), rtruck) + 1
```

```
'Classify the vehicle according to the third length classification, you can change the
numbers!
```

```
ElseIf A(d_array(vehid), length) > 33 Then
```

```

        'Increase movement counter of tapID
        tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), rsemi) =
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), rsemi) + 1

    End If

    'Create subtotal for the right movement
    tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), rtotal) = tapID(A(d_array(vehid),
tap), A(d_array(vehid), strip), rcar) + tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), rtruck) +
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), rsemi)

    'Does the ending edge pixel lie within the through classifying pixel group of that strip
    ElseIf (gblStrips(A(d_array(vehid), strip), Th1) < A(d_array(vehid), sx)) And
(gblStrips(A(d_array(vehid), strip), Th2) > A(d_array(vehid), sx)) Or _
(gblStrips(A(d_array(vehid), strip), Th1) > A(d_array(vehid), sx)) And
(gblStrips(A(d_array(vehid), strip), Th2) < A(d_array(vehid), sx)) Then

        'Average the vehicle summed up vehicle lengths of the A array!
        A(d_array(vehid), length) = (A(d_array(vehid), length)) / (A(d_array(vehid), count))

        'Classify the vehicle according to the first length classification, you can change the
numbers!
        If A(d_array(vehid), length) > 0 And A(d_array(vehid), length) <= 28 Then

            'Increase movement counter of tapID
            tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), tcar) = tapID(A(d_array(vehid),
tap), A(d_array(vehid), strip), tcar) + 1

            'Classify the vehicle according to the second length classification, you can change the
numbers!
            ElseIf A(d_array(vehid), length) > 28 And A(d_array(vehid), length) <= 33 Then

                'Increase movement counter of tapID
                tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), ttruck) =
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), ttruck) + 1

                'Classify the vehicle according to the third length classification, you can change the
numbers!
                ElseIf A(d_array(vehid), length) > 33 Then

                    'Increase movement counter of tapID
                    tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), tsemi) =
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), tsemi) + 1

                    End If

                    'Create subtotal for the through movement
                    tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), ttotal) = tapID(A(d_array(vehid),
tap), A(d_array(vehid), strip), tcar) + tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), ttruck) +
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), tsemi)

                End If

            End If

        End If

```

End If

' Update A array with the new vehID data.

' Is the strip tracking up or left?

If gblStrips(d_array(strip), XX) = 0 Then

A(d_array(vehid), tap) = d_array(tap)

A(d_array(vehid), vehid) = d_array(vehid)

A(d_array(vehid), strip) = d_array(strip)

A(d_array(vehid), track) = d_array(track)

A(d_array(vehid), sx) = d_array(sx)

A(d_array(vehid), sy) = d_array(sy)

A(d_array(vehid), ex) = d_array(ex)

A(d_array(vehid), ey) = d_array(ey)

A(d_array(vehid), length) = d_array(length)

A(d_array(vehid), speed) = d_array(speed)

A(d_array(vehid), count) = 1

' Update A array with the new vehID data.

' Is the strip tracking down or right?

ElseIf gblStrips(d_array(strip), XX) = 1 Then

A(d_array(vehid), tap) = d_array(tap)

A(d_array(vehid), vehid) = d_array(vehid)

A(d_array(vehid), strip) = d_array(strip)

A(d_array(vehid), track) = d_array(track)

A(d_array(vehid), sx) = d_array(ex)

A(d_array(vehid), sy) = d_array(ey)

A(d_array(vehid), ex) = d_array(sx)

A(d_array(vehid), ey) = d_array(sy)

A(d_array(vehid), length) = d_array(length)

A(d_array(vehid), speed) = d_array(speed)

A(d_array(vehid), count) = 1

End If

' It has found a new vehicle for the same strip

ElseIf d_array(track) < A(d_array(vehid), track) Then

' Is the record from a horizontal tracking strip?

If gblStrips(A(d_array(vehid), strip), HV) = 0 Then

' Does the ending edge pixel lie within the classifying pixel group of that strip

If (gblStrips(A(d_array(vehid), strip), E1) < A(d_array(vehid), ey)) And
(gblStrips(A(d_array(vehid), strip), E2) > A(d_array(vehid), ey)) Or _
(gblStrips(A(d_array(vehid), strip), E1) > A(d_array(vehid), ey)) And
(gblStrips(A(d_array(vehid), strip), E2) < A(d_array(vehid), ey)) Then

' Does the ending edge pixel lie within the left classifying pixel group of that strip

```

    If (gblStrips(A(d_array(vehid), strip), L1) < A(d_array(vehid), sy)) And
(gblStrips(A(d_array(vehid), strip), L2) > A(d_array(vehid), sy)) Or _
    (gblStrips(A(d_array(vehid), strip), L1) > A(d_array(vehid), sy)) And
(gblStrips(A(d_array(vehid), strip), L2) < A(d_array(vehid), sy)) Then

    'Average the vehicle summed up vehicle lengths of the A array!
    A(d_array(vehid), length) = (A(d_array(vehid), length)) / (A(d_array(vehid), count))

    'Classify the vehicle according to the first length classification, you can change the
numbers!
    If A(d_array(vehid), length) > 0 And A(d_array(vehid), length) <= 28 Then

        'Increase movement counter of tapID
        tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), lcar) = tapID(A(d_array(vehid),
tap), A(d_array(vehid), strip), lcar) + 1

    'Classify the vehicle according to the second length classification, you can change the
numbers!
    ElseIf A(d_array(vehid), length) > 28 And A(d_array(vehid), length) <= 33 Then

        'Increase movement counter of tapID
        tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), ltruck) =
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), ltruck) + 1

    'Classify the vehicle according to the third length classification, you can change the
numbers!
    ElseIf A(d_array(vehid), length) > 33 Then

        'Increase movement counter of tapID
        tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), lsemi) =
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), lsemi) + 1

    End If

    'Create subtotal for the left movement
    tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), ltotal) = tapID(A(d_array(vehid),
tap), A(d_array(vehid), strip), lcar) + tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), ltruck) +
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), lsemi)

    'Does the ending edge pixel lie within the right classifying pixel group of that strip
    ElseIf (gblStrips(A(d_array(vehid), strip), R1) < A(d_array(vehid), sy)) And
(gblStrips(A(d_array(vehid), strip), R2) > A(d_array(vehid), sy)) Or _
    (gblStrips(A(d_array(vehid), strip), R1) > A(d_array(vehid), sy)) And
(gblStrips(A(d_array(vehid), strip), R2) < A(d_array(vehid), sy)) Then

    'Average the vehicle summed up vehicle lengths of the A array!
    A(d_array(vehid), length) = (A(d_array(vehid), length)) / (A(d_array(vehid), count))

    'Classify the vehicle according to the first length classification, you can change the
numbers!
    If A(d_array(vehid), length) > 0 And A(d_array(vehid), length) <= 28 Then

```

```

    'Increase movement counter of tapID
    tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), rcar) = tapID(A(d_array(vehid),
tap), A(d_array(vehid), strip), rcar) + 1

```

```

    'Classify the vehicle according to the second length classification, you can change the
numbers!

```

```

    ElseIf A(d_array(vehid), length) > 28 And A(d_array(vehid), length) <= 33 Then

```

```

        'Increase movement counter of tapID
        tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), rtruck) =
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), rtruck) + 1

```

```

    'Classify the vehicle according to the third length classification, you can change the
numbers!

```

```

    ElseIf A(d_array(vehid), length) > 33 Then

```

```

        'Increase movement counter of tapID
        tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), rsemi) =
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), rsemi) + 1

```

```

    End If

```

```

    'Create subtotal for the right movement
    tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), rtotal) = tapID(A(d_array(vehid),
tap), A(d_array(vehid), strip), rcar) + tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), rtruck) +
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), rsemi)

```

```

    'Does the ending edge pixel lie within the through classifying pixel group of that strip
    ElseIf (gblStrips(A(d_array(vehid), strip), Th1) < A(d_array(vehid), sy)) And
(gblStrips(A(d_array(vehid), strip), Th2) > A(d_array(vehid), sy)) Or _
(gblStrips(A(d_array(vehid), strip), Th1) > A(d_array(vehid), sy)) And
(gblStrips(A(d_array(vehid), strip), Th2) < A(d_array(vehid), sy)) Then

```

```

    'Average the vehicle summed up vehicle lengths of the A array!

```

```

    A(d_array(vehid), length) = (A(d_array(vehid), length)) / (A(d_array(vehid), count))

```

```

    'Classify the vehicle according to the first length classification, you can change the
numbers!

```

```

    If A(d_array(vehid), length) > 0 And A(d_array(vehid), length) <= 28 Then

```

```

        'Increase movement counter of tapID
        tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), tcar) = tapID(A(d_array(vehid),
tap), A(d_array(vehid), strip), tcar) + 1

```

```

    'Classify the vehicle according to the second length classification, you can change the
numbers!

```

```

    ElseIf A(d_array(vehid), length) > 28 And A(d_array(vehid), length) <= 33 Then

```

```

        'Increase movement counter of tapID
        tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), ttruck) =
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), ttruck) + 1

```

```

'Classify the vehicle according to the third length classification, you can change the
numbers!
ElseIf A(d_array(vehid), length) > 33 Then

    'Increase movement counter of tapID
    tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), tsemi) =
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), tsemi) + 1

    End If

    'Create subtotal for the through movement
    tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), ttotal) = tapID(A(d_array(vehid),
tap), A(d_array(vehid), strip), tcar) + tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), ttruck) +
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), tsemi)

    End If

End If

' Is the record from a vertical tracking strip?
ElseIf gblStrips(A(d_array(vehid), strip), HV) = 1 Then

    'Does the ending edge pixel lie within the classifying pixel group of that strip
    If (gblStrips(A(d_array(vehid), strip), E1) < A(d_array(vehid), ex)) And
(gblStrips(A(d_array(vehid), strip), E2) > A(d_array(vehid), ex)) Or _
    (gblStrips(A(d_array(vehid), strip), E1) > A(d_array(vehid), ex)) And
(gblStrips(A(d_array(vehid), strip), E2) < A(d_array(vehid), ex)) Then

        'Does the ending edge pixel lie within the left classifying pixel group of that strip
        If (gblStrips(A(d_array(vehid), strip), L1) < A(d_array(vehid), sx)) And
(gblStrips(A(d_array(vehid), strip), L2) > A(d_array(vehid), sx)) Or _
        (gblStrips(A(d_array(vehid), strip), L1) > A(d_array(vehid), sx)) And
(gblStrips(A(d_array(vehid), strip), L2) < A(d_array(vehid), sx)) Then
            'Does the last start pixel lie within the left turn area of that strip

            'Average the vehicle summed up vehicle lengths of the A array!
            A(d_array(vehid), length) = (A(d_array(vehid), length)) / (A(d_array(vehid), count))

            'Classify the vehicle according to the first length classification, you can change the
numbers!
            If A(d_array(vehid), length) > 0 And A(d_array(vehid), length) <= 28 Then

                'Increase movement counter of tapID
                tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), lcar) = tapID(A(d_array(vehid),
tap), A(d_array(vehid), strip), lcar) + 1

                'Classify the vehicle according to the second length classification, you can change the
numbers!
                ElseIf A(d_array(vehid), length) > 28 And A(d_array(vehid), length) <= 33 Then

                    'Increase movement counter of tapID
                    tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), ltruck) =
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), ltruck) + 1

```

```

'Classify the vehicle according to the third length classification, you can change the
numbers!
  ElseIf A(d_array(vehid), length) > 33 Then

    'Increase movement counter of tapID
    tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), lsemi) =
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), lsemi) + 1

  End If

'Create subtotal for the left movement
  tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), ltotal) = tapID(A(d_array(vehid),
tap), A(d_array(vehid), strip), lcar) + tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), ltruck) +
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), lsemi)

'Does the ending edge pixel lie within the right classifying pixel group of that strip
  ElseIf (gblStrips(A(d_array(vehid), strip), R1) < A(d_array(vehid), sx)) And
(gblStrips(A(d_array(vehid), strip), R2) > A(d_array(vehid), sx)) Or _
(gblStrips(A(d_array(vehid), strip), R1) > A(d_array(vehid), sx)) And
(gblStrips(A(d_array(vehid), strip), R2) < A(d_array(vehid), sx)) Then

  'Average the vehicle summed up vehicle lengths of the A array!
  A(d_array(vehid), length) = (A(d_array(vehid), length)) / (A(d_array(vehid), count))

'Classify the vehicle according to the first length classification, you can change the
numbers!
  If A(d_array(vehid), length) > 0 And A(d_array(vehid), length) <= 28 Then

    'Increase movement counter of tapID
    tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), rcar) = tapID(A(d_array(vehid),
tap), A(d_array(vehid), strip), rcar) + 1

'Classify the vehicle according to the second length classification, you can change the
numbers!
  ElseIf A(d_array(vehid), length) > 28 And A(d_array(vehid), length) <= 33 Then

    'Increase movement counter of tapID
    tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), rtruck) =
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), rtruck) + 1

'Classify the vehicle according to the third length classification, you can change the
numbers!
  ElseIf A(d_array(vehid), length) > 33 Then

    'Increase movement counter of tapID
    tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), rsemi) =
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), rsemi) + 1

  End If

'Create subtotal for the right movement

```

```

        tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), rtotal) = tapID(A(d_array(vehid),
tap), A(d_array(vehid), strip), rcar) + tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), rtruck) +
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), rsemi)

```

```

        'Does the ending edge pixel lie within the through classifying pixel group of that strip
        ElseIf (gblStrips(A(d_array(vehid), strip), Th1) < A(d_array(vehid), sx)) And
(gblStrips(A(d_array(vehid), strip), Th2) > A(d_array(vehid), sx)) Or _
        (gblStrips(A(d_array(vehid), strip), Th1) > A(d_array(vehid), sx)) And
(gblStrips(A(d_array(vehid), strip), Th2) < A(d_array(vehid), sx)) Then

```

```

        'Average the vehicle summed up vehicle lengths of the A array!

```

```

        A(d_array(vehid), length) = (A(d_array(vehid), length)) / (A(d_array(vehid), count))

```

```

        'Classify the vehicle according to the first length classification, you can change the
numbers!

```

```

        If A(d_array(vehid), length) > 0 And A(d_array(vehid), length) <= 28 Then

```

```

            'Increase movement counter of tapID

```

```

            tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), tcar) = tapID(A(d_array(vehid),
tap), A(d_array(vehid), strip), tcar) + 1

```

```

        'Classify the vehicle according to the second length classification, you can change the
numbers!

```

```

        ElseIf A(d_array(vehid), length) > 28 And A(d_array(vehid), length) <= 33 Then

```

```

            'Increase movement counter of tapID

```

```

            tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), ttruck) =
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), ttruck) + 1

```

```

        'Classify the vehicle according to the third length classification, you can change the
numbers!

```

```

        ElseIf A(d_array(vehid), length) > 33 Then

```

```

            'Increase movement counter of tapID

```

```

            tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), tsemi) =
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), tsemi) + 1

```

```

        End If

```

```

        'Create subtotal for the through movement

```

```

        tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), ttotal) = tapID(A(d_array(vehid),
tap), A(d_array(vehid), strip), tcar) + tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), ttruck) +
tapID(A(d_array(vehid), tap), A(d_array(vehid), strip), tsemi)

```

```

    End If

```

```

End If

```

```

End If

```

```

' Update A array with the new vehID data.

```

```

' Is the strip tracking up or left?
If gblStrips(d_array(strip), XX) = 0 Then

    A(d_array(vehid), tap) = d_array(tap)
    A(d_array(vehid), vehid) = d_array(vehid)
    A(d_array(vehid), strip) = d_array(strip)
    A(d_array(vehid), track) = d_array(track)
    A(d_array(vehid), sx) = d_array(sx)
    A(d_array(vehid), sy) = d_array(sy)
    A(d_array(vehid), ex) = d_array(ex)
    A(d_array(vehid), ey) = d_array(ey)
    A(d_array(vehid), length) = d_array(length)
    A(d_array(vehid), speed) = d_array(speed)
    A(d_array(vehid), count) = 1

' Update A array with the new vehID data.
' Is the strip tracking down or right?
ElseIf gblStrips(d_array(strip), XX) = 1 Then

    A(d_array(vehid), tap) = d_array(tap)
    A(d_array(vehid), vehid) = d_array(vehid)
    A(d_array(vehid), strip) = d_array(strip)
    A(d_array(vehid), track) = d_array(track)
    A(d_array(vehid), sx) = d_array(ex)
    A(d_array(vehid), sy) = d_array(ey)
    A(d_array(vehid), ex) = d_array(sx)
    A(d_array(vehid), ey) = d_array(sy)
    A(d_array(vehid), length) = d_array(length)
    A(d_array(vehid), speed) = d_array(speed)
    A(d_array(vehid), count) = 1

End If

End If

'Read next record in the database.
rsData.MoveNext

Loop

End With

'Close the recordset and database.
rsData.Close

Set db = Nothing

' Assign turn movements to the remaining vehID's, when there is no more new ones
For i = 0 To 31

' Is the record from a horizontal tracking strip?
If gblStrips(A(i, strip), HV) = 0 Then

    'Does the ending edge pixel lie within the classifying pixel group of that strip

```

If (gblStrips(A(i, strip), E1) < A(i, ey)) And (gblStrips(A(i, strip), E2) > A(i, ey)) Or _
(gblStrips(A(i, strip), E1) > A(i, ey)) And (gblStrips(A(i, strip), E2) < A(i, ey)) Then

'Does the ending edge pixel lie within the left classifying pixel group of that strip
If (gblStrips(A(i, strip), L1) < A(i, sy)) And (gblStrips(A(i, strip), L2) > A(i, sy)) Or _
(gblStrips(A(i, strip), L1) > A(i, sy)) And (gblStrips(A(i, strip), L2) < A(i, sy)) Then
'Does the last start pixel lie within the left turn area of that strip

'Average the vehicle summed up vehicle lengths of the A array!
A(i, length) = (A(i, length)) / (A(i, count))

'Classify the vehicle according to the first length classification, you can change the numbers!
If A(i, length) > 0 And A(i, length) <= 28 Then

'Increase movement counter of tapID
tapID(A(i, tap), A(i, strip), lcar) = tapID(A(i, tap), A(i, strip), lcar) + 1

'Classify the vehicle according to the second length classification, you can change the numbers!
ElseIf A(d_array(vehid), length) > 28 And A(d_array(vehid), length) <= 33 Then

'Increase movement counter of tapID
tapID(A(i, tap), A(i, strip), ltruck) = tapID(A(i, tap), A(i, strip), ltruck) + 1

'Classify the vehicle according to the third length classification, you can change the numbers!
ElseIf A(d_array(vehid), length) > 33 Then

'Increase movement counter of tapID
tapID(A(i, tap), A(i, strip), lsemi) = tapID(A(i, tap), A(i, strip), lsemi) + 1

End If

'Create subtotal for the left movement
tapID(A(i, tap), A(i, strip), ltotal) = tapID(A(i, tap), A(i, strip), lcar) + tapID(A(i, tap), A(i, strip),
ltruck) + tapID(A(i, tap), A(i, strip), lsemi)

'Does the ending edge pixel lie within the right classifying pixel group of that strip
ElseIf (gblStrips(A(i, strip), R1) < A(i, sy)) And (gblStrips(A(i, strip), R2) > A(i, sy)) Or _
(gblStrips(A(i, strip), R1) > A(i, sy)) And (gblStrips(A(i, strip), R2) < A(i, sy)) Then

'Average the vehicle summed up vehicle lengths of the A array!
A(i, length) = (A(i, length)) / (A(i, count))

'Classify the vehicle according to the first length classification, you can change the numbers!
If A(i, length) > 0 And A(i, length) <= 28 Then

'Increase movement counter of tapID
tapID(A(i, tap), A(i, strip), rcar) = tapID(A(i, tap), A(i, strip), rcar) + 1

'Classify the vehicle according to the second length classification, you can change the numbers!
ElseIf A(d_array(vehid), length) > 28 And A(d_array(vehid), length) <= 33 Then

'Increase movement counter of tapID
tapID(A(i, tap), A(i, strip), rtruck) = tapID(A(i, tap), A(i, strip), rtruck) + 1

```

'Classify the vehicle according to the third length classification, you can change the numbers!
ElseIf A(d_array(vehid), length) > 33 Then

    'Increase movement counter of tapID
    tapID(A(i, tap), A(i, strip), rsemi) = tapID(A(i, tap), A(i, strip), rsemi) + 1

End If

'Create subtotal for the right movement
tapID(A(i, tap), A(i, strip), rtotal) = tapID(A(i, tap), A(i, strip), rcar) + tapID(A(i, tap), A(i, strip),
rtruck) + tapID(A(i, tap), A(i, strip), rsemi)

'Does the ending edge pixel lie within the through classifying pixel group of that strip
ElseIf (gblStrips(A(i, strip), Th1) < A(i, sy)) And (gblStrips(A(i, strip), Th2) > A(i, sy)) Or _
(gblStrips(A(i, strip), Th1) > A(i, sy)) And (gblStrips(A(i, strip), Th2) < A(i, sy)) Then

    'Average the vehicle summed up vehicle lengths of the A array!
    A(i, length) = (A(i, length)) / (A(i, count))

    'Increase movement counter of tapID
    If A(i, length) > 0 And A(i, length) <= 28 Then

        'Classify the vehicle according to the first length classification, you can change the numbers!
        tapID(A(i, tap), A(i, strip), tcar) = tapID(A(i, tap), A(i, strip), tcar) + 1

        'Increase movement counter of tapID
        ElseIf A(i, length) > 28 And A(d_array(vehid), length) <= 33 Then

            'Classify the vehicle according to the second length classification, you can change the
numbers!
            tapID(A(i, tap), A(i, strip), ttruck) = tapID(A(i, tap), A(i, strip), ttruck) + 1

            'Increase movement counter of tapID
            ElseIf A(i, length) > 33 Then

                'Classify the vehicle according to the third length classification, you can change the numbers!
                tapID(A(i, tap), A(i, strip), tsemi) = tapID(A(i, tap), A(i, strip), tsemi) + 1

            End If

            'Create subtotal for the through movement
            tapID(A(i, tap), A(i, strip), tttotal) = tapID(A(i, tap), A(i, strip), tcar) + tapID(A(i, tap), A(i, strip),
ttruck) + tapID(A(i, tap), A(i, strip), tsemi)

        End If

    End If

' Is the record from a vertical tracking strip?

```

ElseIf gblStrips(A(i, strip), HV) = 1 Then

'Does the ending edge pixel lie within the classifying pixel group of that strip
If (gblStrips(A(i, strip), E1) < A(i, ex)) And (gblStrips(A(i, strip), E2) > A(i, ex)) Or _
(gblStrips(A(i, strip), E1) > A(i, ex)) And (gblStrips(A(i, strip), E2) < A(i, ex)) Then

'Does the ending edge pixel lie within the left classifying pixel group of that strip
If (gblStrips(A(i, strip), L1) < A(i, sx)) And (gblStrips(A(i, strip), L2) > A(i, sx)) Or _
(gblStrips(A(i, strip), L1) > A(i, sx)) And (gblStrips(A(i, strip), L2) < A(i, sx)) Then

'Average the vehicle summed up vehicle lengths of the A array!
A(i, length) = (A(i, length)) / (A(i, count))

'Classify the vehicle according to the first length classification, you can change the numbers!
If A(i, length) > 0 And A(i, length) <= 28 Then

'Increase movement counter of tapID
tapID(A(i, tap), A(i, strip), lcar) = tapID(A(i, tap), A(i, strip), lcar) + 1

'Classify the vehicle according to the second length classification, you can change the numbers!
ElseIf A(i, length) > 28 And A(i, length) <= 33 Then

'Increase movement counter of tapID
tapID(A(i, tap), A(i, strip), ltruck) = tapID(A(i, tap), A(i, strip), ltruck) + 1

'Classify the vehicle according to the third length classification, you can change the numbers!
ElseIf A(i, length) > 33 Then

'Increase movement counter of tapID
tapID(A(i, tap), A(i, strip), lsemi) = tapID(A(i, tap), A(i, strip), lsemi) + 1

End If

'Create subtotal for the left movement
tapID(A(i, tap), A(i, strip), ltotal) = tapID(A(i, tap), A(i, strip), lcar) + tapID(A(i, tap), A(i, strip),
ltruck) + tapID(A(i, tap), A(i, strip), lsemi)

'Does the ending edge pixel lie within the right classifying pixel group of that strip
ElseIf (gblStrips(A(i, strip), R1) < A(i, sx)) And (gblStrips(A(i, strip), R2) > A(i, sx)) Or _
(gblStrips(A(i, strip), R1) > A(i, sx)) And (gblStrips(A(i, strip), R2) < A(i, sx)) Then

'Average the vehicle summed up vehicle lengths of the A array!
A(i, length) = (A(i, length)) / (A(i, count))

'Classify the vehicle according to the first length classification, you can change the numbers!
If A(i, length) > 0 And A(i, length) <= 28 Then

'Increase movement counter of tapID
tapID(A(i, tap), A(i, strip), rcar) = tapID(A(i, tap), A(i, strip), rcar) + 1

'Classify the vehicle according to the second length classification, you can change the numbers!
ElseIf A(i, length) > 28 And A(i, length) <= 33 Then

```

'Increase movement counter of tapID
tapID(A(i, tap), A(i, strip), rtruck) = tapID(A(i, tap), A(i, strip), rtruck) + 1

'Classify the vehicle according to the third length classification, you can change the numbers!
ElseIf A(i, length) > 33 Then

    'Increase movement counter of tapID
    tapID(A(i, tap), A(i, strip), rsemi) = tapID(A(i, tap), A(i, strip), rsemi) + 1

End If

'Create subtotal for the right movement
tapID(A(i, tap), A(i, strip), rtotal) = tapID(A(i, tap), A(i, strip), rcar) + tapID(A(i, tap), A(i, strip),
rtruck) + tapID(A(i, tap), A(i, strip), rsemi)

'Does the ending edge pixel lie within the through classifying pixel group of that strip
ElseIf (gblStrips(A(i, strip), Th1) < A(i, sx)) And (gblStrips(A(i, strip), Th2) > A(i, sx)) Or _
(gblStrips(A(i, strip), Th1) > A(i, sx)) And (gblStrips(A(i, strip), Th2) < A(i, sx)) Then

'Average the vehicle summed up vehicle lengths of the A array!
A(i, length) = (A(i, length)) / (A(i, count))

'Classify the vehicle according to the first length classification, you can change the numbers!
If A(i, length) > 0 And A(i, length) <= 28 Then

    'Increase movement counter of tapID
    tapID(A(i, tap), A(i, strip), tcar) = tapID(A(i, tap), A(i, strip), tcar) + 1

'Classify the vehicle according to the second length classification, you can change the numbers!
ElseIf A(i, length) > 28 And A(i, length) <= 33 Then

    'Increase movement counter of tapID
    tapID(A(i, tap), A(i, strip), ttruck) = tapID(A(i, tap), A(i, strip), ttruck) + 1

'Classify the vehicle according to the third length classification, you can change the numbers!
ElseIf A(d_array(vehid), length) > 33 Then

    'Increase movement counter of tapID
    tapID(A(i, tap), A(i, strip), tsemi) = tapID(A(i, tap), A(i, strip), tsemi) + 1

End If

'Create subtotal for the through movement
tapID(A(i, tap), A(i, strip), tttotal) = tapID(A(i, tap), A(i, strip), tcar) + tapID(A(i, tap), A(i, strip),
ttruck) + tapID(A(i, tap), A(i, strip), tsemi)

End If

End If

End If

```

Next i

frmResult.Show

End Sub

'After this Algorithm over, all results are stored in the tapID(8, 4, 11).
'One must display the results of the array to see the answers.