1-1-2008

A mathematical model for computerized car crash detection using computer vision techniques

Dawn Marie Strianese
University of Nevada, Las Vegas

A MATHEMATICAL MODEL FOR COMPUTERIZED CAR CRASH DETECTION USING COMPUTER VISION TECHNIQUES

by

Dawn Marie Strianese

Bachelor of Science in Computer Science
Westminster College
Salt Lake City, Utah
1994

A thesis submitted in partial fulfillment
of the requirements for the

Master of Science in Computer Science
School of Computer Science
University of Nevada, Las Vegas

Graduate College
University of Nevada, Las Vegas
May 2008

UNLV
UNIVERSITY OF NEVADA LAS VEGAS

Thesis Approval
The Graduate College
University of Nevada, Las Vegas

MARCH 24TH, 2008

The Thesis prepared by

DAWN MARIE STRIANESE

Entitled

A MATHEMATICAL MODEL FOR COMPUTARIZED CAR CRASH DETECTION USING COMPUTER VISION TECHNIQUES

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

MASTER OF SCIENCE IN COMPUTER SCIENCE

Examination Committee Chair

Dean of the Graduate College

Glenn T. Minor
Examination Committee Member

Examination Committee Member

3/24/08
Graduate College Faculty Representative

ABSTRACT

A Mathematical Model for Computerized Car Crash Detection using Computer Vision Techniques

by

Dawn Marie Strianese

Dr. Evangelos Yfantis, Examination Committee Chair
Professor of Computer Science
University of Nevada, Las Vegas

My proposed approach to the automatic detection of traffic accidents in a signalized intersection is presented here. In this method, a digital camera is strategically placed to view the entire intersection. The images are captured, processed and analyzed for the presence of vehicles and pedestrians in the proposed detection zones. Those images are further processed to detect if an accident has occurred.

The mathematical model presented is a Poisson distribution that predicts the number of accidents in an intersection per week, which can be used as approximations for modeling the crash process. We believe that the crash process can be modeled by using a two-state method, which implies that the intersection is in one of two states: clear (no accident) or obstructed (accident). We can then incorporate a rule-based AI system, which will help us in identifying that a crash has taken or will possibly take place.

We have modeled the intersection as a service facility, which processes vehicles in a relatively small amount of time. A traffic accident is then perceived as an interruption of that service.

TABLE OF CONTENTS

ABSTRACT iii

LIST OF TABLES vi

LIST OF FIGURES viii

LIST OF EQUATIONS x

ACKNOWLEDGMENTS xi

CHAPTER 1 INTRODUCTION 1
What Is a Car Crash? 1
Disregard for the Rules of the Road 1
Computer Vision 4
Motivation 8
Disclaimer 9

CHAPTER 2 BACKGROUND / RELATED WORK 10
A Brief Introduction to Statistics 10
A Brief Overview of a Visual Surveillance System 12
A Brief Overview of Computer Vision 13
Background 15
Common Problems 18
Image Acquisition and Pre-Processing 19
Histogram Equalization 19
Noise Suppression 19
What is RGB color? 20
Moving Object Detection (Motion Segmentation) 21
Background Subtraction 21
Statistical Methods for Background Subtraction 23
Temporal Differencing for Background Subtraction 23
Image Segmentation 24
Thresholding 25
Binary Images 25
Morphology 26
Smoothing 30
Convolutions 30
Gaussian Filter 31
Median Filter 31

Edge Detection 32
Canny 32
Sobel 33
Laplace 34
Connected Components (Object Labeling) 35
Moving Object Classification 36
Shape-Based Classification 37
Motion-Based Classification 38
Color-Based Classification 39
Object Recognition 39
Shadow Detection and Suppression 40
Finding Vehicles by Appearance 43
Tracking Moving Objects 43
Optical Flow / Motion Estimation 44
The Continuously Adaptive Mean Shift Algorithm (Camshift) 44
Kalman Filter 45
Occlusion 45
Challenges 46
Evaluation 46

CHAPTER 3 METHODOLOGY 47
Modeling Traffic Flow at a Signalized Intersection 47
Review of the Crash Statistics 53
Implementation and Specifics 58
Moving Object Detection (Motion Segmentation) 58
Adaptive Background Subtraction Model 58
Noise Suppression 59
Thresholding, Erode, Dilation 59
Edge Detection 60
Shadow Detection and Suppression 60
Object Classification 62
Tracking Moving Objects 64
Occlusion 64

CHAPTER 4 RESULTS AND CONCLUSIONS 65
Future Work 65

APPENDIX A DATA ANALYSIS 67

APPENDIX B SUMMARY OF EDGE DETECTORS 101

APPENDIX C EXPERIMENTAL PROGRAM RESULTS 102

REFERENCES... 104

VITA............. 113

LIST OF TABLES

Table 1 Statewide crash statistics for the state of Nevada over the last 5 years [80] .. 53
Table 2 Ann and Simmons weekly total of accidents for 2004 55
Table 3 The maximum number of accidents per week was 2 for Ann and Simmons . 56
Table 4 There were a total of 44 weeks without an accident for Ann and Simmons .. 56
Table 5 Poisson distribution for Ann and Simmons 57
Table 6 Total number of accidents by week for one year for Las Vegas Blvd and Flamingo Road. 67
Table 7 There were a total of 7 weeks without an accident for Las Vegas Blvd and Flamingo Road 67
Table 8 The maximum number of accidents reported in one week was 7 for Las Vegas Blvd and Flamingo Road 68
Table 9 Poisson distribution for Las Vegas Blvd and Flamingo Road 68
Table 10 Total number of accidents by week for one year for Main and Charleston.... 70
Table 11 There were a total of 28 weeks without an accident for Main & Charleston . 70
Table 12 The maximum number of accidents reported in one week was 3 for Main and Charleston 70
Table 13 Poisson distribution for Main and Charleston 71
Table 14 Total number of accidents by week for 1 year for Maryland & Tropicana 72
Table 15 There were a total of 15 weeks without an accident for Maryland & Tropicana 72
Table 16 The maximum number of accidents reported in one week was 4 for Maryland and Tropicana.. 72
Table 17 Poisson distribution for Maryland & Tropicana 73
Table 18 Total number of accidents by week for one year for MLK and Cheyenne..... 74
Table 19 There were a total of 17 weeks without an accident for MLK and Cheyenne 74
Table 20 The maximum number of accidents reported in one week was 4 for MLK and Cheyenne.. 74
Table 21 Poisson distribution for MLK and Cheyenne 75
Table 22 Total number of accidents by week for one year for MLK and Craig............ 77
Table 23 There were a total of 14 weeks without an accident for MLK and Craig....... 77
Table 24 The maximum number of accidents reported in one week was 5 for MLK and Craig... 77
Table 25 Poisson distribution for Martin Luther King and Craig................................. 78
Table 26 Total number of accidents by week for one year for Sahara and Durango..... 80
Table 27 There were a total of 15 weeks without an accident for Sahara & Durango .. 80
Table 28 The maximum number of accidents reported in one week was 4 for Sahara and Durango.. 80
Table 29 Poisson distribution for Sahara and Durango 81

Table 30 Total number of accidents by week for one year for Warm Springs and Rainbow 83
Table 31 There were a total of 28 weeks without an accident for Warm Springs and Rainbow 83
Table 32 The maximum number of accidents reported in one week was 3 for Warm Springs and Rainbow 83
Table 33 Poisson distribution for Warm Springs and Rainbow 84
Table 34 Total number of accidents by week for one year for Jones and Sahara 86
Table 35 There were a total of 16 weeks without an accident for Jones and Sahara 86
Table 36 The maximum number of accidents reported in one week was four for Jones and Sahara 86
Table 37 Poisson distribution for Jones and Sahara 87
Table 38 Total number of accidents by week for one year for Flamingo & Maryland . 89
Table 39 There were a total of five weeks without an accident for Flamingo and Maryland 89
Table 40 The maximum number of accidents reported in one week was 7 for Flamingo and Maryland 89
Table 41 Poisson distribution for Flamingo and Maryland 90
Table 42 Total number of accidents by week for one year for Cheyenne and Rancho . 92
Table 43 There were a total of 28 weeks without an accident for Cheyenne and Rancho 92
Table 44 The maximum number of accidents reported in 1 week was three for Cheyenne and Rancho 92
Table 45 Poisson distribution for Cheyenne and Rancho 93
Table 46 Total number of accidents by week for one year for Buffalo and Lone Mountain 95
Table 47 There were a total of 44 weeks without an accident for Buffalo and Lone Mountain 95
Table 48 The maximum number of accidents reported in one week was 2 for Buffalo and Lone Mountain 95
Table 49 Poisson distribution for Buffalo and Lone Mountain 96
Table 50 Total number of accidents by week for one year for Lamb and Craig 98
Table 51 There were a total of 34 weeks without an accident for Lamb and Craig 98
Table 52 The maximum number of accidents reported in one week was three for Lamb and Craig 98
Table 53 Poisson distribution for Lamb and Craig 99
Table 54 Summary of Edge Detectors 101

LIST OF FIGURES

Figure 1	An example of a Poisson distribution graph. The horizontal axis is the index x. The function is defined only at integer values of x	11
Figure 2	Pixel image array	15
Figure 3	The color spectrum	20
Figure 4	A 4 and 8 connected grid example	27
Figure 5	A 4 and 8 adjacent example	27
Figure 6	The effect of erosion using a 3x3 square structuring element	29
Figure 7	The effect of dilation using a 3x3 square structuring element	29
Figure 8	The effect of opening using a 3x3 square structuring element	29
Figure 9	The effect of closing using a 3x3 square structuring element	30
Figure 10	Blurring convolution mask	30
Figure 11	Gaussian filter	31
Figure 12	Sobel convolution kernels	33
Figure 13	Sample edge	34
Figure 14	The first derivative, which is the gradient	34
Figure 15	The second derivative	35
Figure 16	The Laplace operator	35
Figure 17	Shadow parts	41
Figure 18	A typical intersection model	47
Figure 19	Active / Inactive zones	49
Figure 20	Ingress and Egress zones	50
Figure 21	Common traffic paths	51
Figure 22	Poisson chart for Ann and Simmons	57
Figure 23	Example of contour by erosion	60
Figure 24	Original Image	61
Figure 25	Image with detected shadows before truncation	61
Figure 26	Image with shadows truncated	62
Figure 27	Illustration of finding the shortest line	63
Figure 28	Poisson chart for Las Vegas Blvd and Flamingo Road	69
Figure 29	Poisson chart for Main and Charleston	71
Figure 30	Poisson chart for Maryland and Tropicana	73
Figure 31	Poisson chart for Martin Luther King and Cheyenne	76
Figure 32	Poisson chart for Martin Luther King and Craig	79
Figure 33	Poisson chart for Sahara and Durango	82
Figure 34	Poisson chart for Warm Springs and Rainbow	85
Figure 35	Poisson chart for Jones and Sahara	88
Figure 36	Poisson chart for Flamingo and Maryland	91
Figure 37	Poisson chart for Cheyenne and Rancho	94
Figure 38	Poisson chart for Buffalo and Lone Mountain	97

Figure 39 Poisson chart for Lamb and Craig 100
Figure 40 Example of using Optical Flow to detect 'backward' movement in the intersection 102
Figure 41 Example of using 'reference frame' to detect objects that have not left the intersection and have stopped moving 103

LIST OF EQUATIONS

Equation 1	Poisson distribution function	10
Equation 2	Standard conversion of an RGB image to grayscale	21
Equation 3	Formal definition of thresholding an image	25
Equation 4	Formal definition of a binary image	26
Equation 5	Standard mathematical morphological operations where A is the image and B is the structuring element	28
Equation 6	Function used to calculate the Poisson distribution	54
Equation 7	Inter-arrival time	55
Equation 8	Adaptive background subtraction function	58

ACKNOWLEDGMENTS

It is a pleasure to thank the many people who made this thesis possible.

I would like to thank my advisor, Dr. Evangelos Yfantis, for his guidance and support. Thank you to my thesis examination committee, Dr. John Minor, Dr. Laxmi Gewali and Dr. Mohamed Kaseko for their time and willingness to participate. A special thanks to Dr. Mohamed Kaseko and his department for providing me all of the necessary statistical data used in this thesis.

This work would not have been possible without the support and encouragement of my family, especially my mother. I would also like to thank Frank Bayer, for proof reading my thesis, time and time again, to make sure that I had it just right.

CHAPTER 1

INTRODUCTION

According to current available data, intersection accidents in the state of Nevada over the last five years comprise approximately 32% of the total number of reported accidents [80]. If we remove accidents reported by the Nevada Highway Patrol, then the number is almost 50%.

Despite the many advances in roadway design, traffic engineering, traffic control devices, automobile manufacturing and police enforcement technologies, we all still suffer from the ever-increasing problem of traffic accidents.

What Is a Car Crash?

A car crash is an accident resulting from the collision of two or more vehicles, which causes a disruption to the normal traffic flow.

In 1949, the United Nations set forth a protocol on road signs and signals, which states that Red indicates stop [81]. The Manual on Uniform Traffic Control Devices or MUTCD lists standards that are defined nationwide governing the installation, usage and maintenance of traffic control devices on all streets and highways. This manual was prepared by a national committee with the National Highway Administration, and comprises standards that pertain to all roads and freeways. It was concluded that simply adding more traffic control devices would reduce traffic accidents was a myth [82].

Disregard for the Rules of the Road

What are the current traffic laws that apply to an intersection?

The rules of the road are traffic laws and driving practices that equate to safe driving. The current traffic laws for the state of Nevada are outlined in the Nevada's Driver's Handbook from the Department of Motor Vehicles (DMV) and a complete list is documented in the Nevada Revised Statutes which are publicly available [87].

These rules include:

- Signs and signals
- Right of way
- Controlling speed
- Stopping
- Signaling, turning, lane changes and passing

Some common traffic violations that occur in an intersection are:

- Running a red light
- Failure to yield the right-of-way
- Changing lanes in an intersection
- Illegal turns
- Blocking an intersection
- Stopping in an intersection
- Failure to yield to an emergency vehicle
- Failure to yield for pedestrians
- Turning from the wrong lane
- Turning into the wrong lane

- Speeding
- Failure to use turn signals
- Failure to turn in a turn only lane

Potential Problems with the Crash Data

Crashes are random events, therefore it is important to note that the number / rate of crashes at a location are not an accurate gauge.

Not all traffic accidents are reported to the police. Some common reasons why an accident would not be reported are:

- Driving without a license, insurance, or registration
- Drunk driving
- Fear of insurance rate increase
- Possession of something illegal such as a firearm, drugs, etc
- The vehicle involved is stolen or not street legal
- The driver is a wanted criminal

In the state of Nevada, all accidents which involve injuries or damages that exceed $750 must be reported. If a police officer does not look into the accident, all of the parties involved must file an accident report form (SR-1) within 10 days [87].

Crash reports may contain errors such as

- Typos
- Misspellings
- Failure to observe all circumstances due to poor lighting or weather
- Misinterpretation of the scene

- Incorrect or misleading reports from those involved

Crashes are normally the consequence of a chain of events that involve the interaction of three factors: the road, the vehicle, and the driver. Some of these factors are:

- Weather – rain, snow, black ice
- Road construction
- The road is in need of repair (lane markings are not clear, potholes, etc)
- Foreign objects obstructing the road like animals, debris, etc
- Vehicle is in poor mechanical condition
- Tire blowout
- Vehicle lights do not work
- Driver falls asleep at the wheel
- Driver is distracted due to a cell phone, eating, shaving, reading a map, lost, etc
- Inexperienced driver
- Driver has a medical condition or experiences a medical problem
- Driver has a visual impairment (forgot glasses, sunglasses, etc)
- Driver is unfamiliar with the vehicle (rental car, do not know how to drive a manual transmission, etc)

Computer Vision

As part of this thesis, I will present an intelligent visual surveillance system, using computer vision techniques, moving object detection, tracking and classification methods. The proposed system will operate on color video images taken from a static, stationary camera. This approach can easily be modified to make use of one or more cameras. This approach can also be adapted to a non-signalized intersection, such as a four-way stop.

The objective of any computer vision system is to analyze and interpret digital, visual data and use that information to complete a predefined task. Object recognition or classification of objects into known classes, is a vital part of computer vision.

The task of object recognition is a complex one. How do we interpret information about two-dimensional images in a three-dimensional world? Recognition implies awareness and previous knowledge about something. How can someone recognize something unless they know what they are looking for? The human visual system can recognize many different kinds of objects easily; however, visual recognition is generally a difficult task for a computer [38].

The real world that we see and touch is composed, predominantly, of concrete three-dimensional objects. When someone is given an object that they have never seen before, they typically collect and analyze information about the object by looking at it from many different positions, and subsequently identify it.

"A picture is worth a thousand words" [15] is a familiar proverb that refers to the idea that complex stories can be told with a single still image, or that an image may be more influential than a substantial amount of text. I believe this phrase sums up the importance and complexity of computer vision in general.

The creation of an elegant system requires fast, reliable and robust algorithms for moving object detection, classification, tracking and analysis. Automatic detection and recognition of moving objects is of primary importance for video surveillance applications. Automated systems deal with real-time observations of vehicles and people within a busy and sometimes cluttered outdoor environment and must be able to identify and track objects moving in its field of vision.

My approach is based on extracting objects in the form of blobs in the scene using a motion segmentation method, tracking these objects while they appear in the region of interest and classifying these objects into pedestrians and vehicles. This approach makes use of a single, static camera mounted above the scene with a clear view of the region of interest.

Image acquisition and pre-processing are the very first steps. Image acquisition is the starting point because it is the reference image or video stream for the algorithm. A digital image is produced by one or several image capture devices, which vary by type and application. Depending on the type of the input device, the output image created is either a 2D image or an image sequence. The value of each pixel usually corresponds to light intensity (gray images or color images), but they can also be related to a variety of physical measures, such as depth, absorption or reflectance. Prior to a computer vision method being applied to the image, so that specific piece(s) of information can be extracted, it is usually necessary to process the image in order to confirm that it satisfies certain assumptions implied by the method, such as contrast enhancement and noise reduction. In an attempt to simplify the final classification, which is more accurate, the process of smoothing, enhancing, filtering and cleaning up of the digital image is completed.

Moving object detection or motion segmentation is the next step in the analysis of a video stream. During this process, a choice is made about which regions in the image are relevant for further processing. It handles the segmentation of moving objects from the stationary background objects. This not only creates a focus for higher level processing, but also decreases the computational time significantly. Commonly used techniques for moving object detection are background subtraction, statistical models, temporal differencing and optical flow. Due to dynamic outdoor environmental conditions such as illumination changes, shadows and tree branches moving in the wind, object segmentation is a difficult problem that needs to be handled properly for a reliable and robust system. Even though background subtraction techniques perform well at extracting most of the relevant pixels of moving regions even when they stop, they can still be sensitive to dynamic changes; for instance, stationary objects uncovering the background (i.e. a parked car moves) or sudden illumination changes.

Before blobs can be treated as objects, the blobs need to be identified from the output of the motion segmentation step. Blobs can be considered a spatially coherent group of pixels which are clustered together that represent an object. Categorizing the pixels in an image as belonging to one of many distinct regions is an image segmentation technique commonly known as blob extraction. Blob extraction is executed on the resulting binary image from the thresholding step.

Moving object classification is the next step in the analysis of an incoming video stream. In this process the applicable shape information inherent in a pattern is computed making it easier to classify the pattern in a later function. This process classifies all of the detected objects into preset classes such as pedestrian, vehicle, other, etc. It is necessary to differentiate objects from each other in order to track them reliably. Presently, there are two major approaches in moving object classification, the shape-based and motion-based methods. Motion-based techniques use the temporal features of objects for the classification step while shape-based techniques use an object's 2-dimensional spatial information. A feature is a useful and consistent characteristic in an object or blob, such as a geometric shape.

Tracking the moving objects is the last step in the analysis of an incoming video stream. It can be defined as the formation of temporal correspondence between detected objects from frame to frame. The result of this step is commonly used to support and improve motion segmentation, object classification and other processing.

There are many advantages to video surveillance and detection. Installation of a camera would have a minimal impact on the traffic flow, since the proposed location of the camera is outside of the intersection.

Some of the disadvantages of using automatic video detection include improper segmentation due to poor weather conditions, camera vibrations from high winds and dirt collecting on the camera lens obstructing the photographic record.

Due to NRS regulation 484.910, which prohibits photographic devices for traffic law enforcement, the recording of live traffic data in the state of Nevada is prevented. As a result, data was simulated and used by my experimental program.

Motivation

In the dozens of science fiction movies that I have enjoyed over the years, I have always been curious to know exactly how a machine, or a robot, can see. That is, how can a computer detect and recognize objects that are unfamiliar to it? So how do you teach a machine how to interpret what it "sees" and / or how do you teach a machine how to act or react on what it “sees” or thinks that it sees? These are just a few of the reasons why I choose Computer Science.

Disclaimer

Since this paper is an entire application (all encompassing), a comparison of individual techniques for each part is beyond the scope of this work. All processing algorithms, such as background subtraction, shadow detection and removal, etc, currently available are not included - such research is left to the reader.

CHAPTER 2

BACKGROUND / RELATED WORK

A Brief Introduction to Statistics

With the help of the Bureau of Transportation Statistics (BTS) and UNLV Transportation Research Center, I have collected data about accidents in the state of Nevada taking place at intersections in the Las Vegas area. An analysis of the data has yielded a Poisson statistical model, when the accidents are grouped into a weekly time period. This will aid us in identifying intersections with a high volume of accidents and allow us to predict when the next accident will likely occur.

The Poisson distribution is a discrete probability distribution that states the probability of a number of events occurring in a fixed time period when these events occur with a known average rate and are independent of each other.

The distribution was discovered by Simeon-Denis Poisson and published in 1838 [75]. The work focused on random variables N that count a number of separate occurrences, or arrivals, that take place during an interval of a given length of time. If the expected number of occurrences in this interval is λ, then the probability that there are exactly k occurrences, where k is a positive integer, is equal to:

$$p(x, \lambda) = \frac{e^{-\lambda}\lambda^x}{x!} \qquad \text{for } x = 0, 1, 2, \cdots$$

Equation 1 Poisson distribution function

where e is the natural logarithm, x is the number of occurrences of an event with the probability which is represented by the function, λ is the parameter which indicates the average number of events in the given time interval. The parameter λ is not only the mean number of occurrences (k), but also its variance.

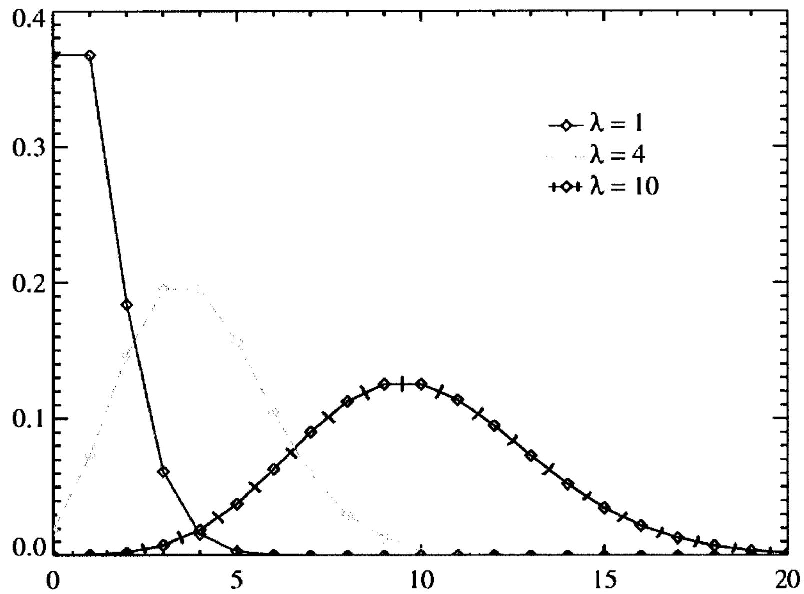


Figure 1 An example of a Poisson distribution graph. The horizontal axis is the index x. The function is defined only at integer values of x.

Traffic accidents are events that are normally independent of each other. An intersection is where the events take place. Traffic accidents occur every day. An intersection can be seen as the convergence of two or more roads.

We believe that the vehicle crash process can be modeled by using a two-state method, which implies that the intersection exists in one of two states, clear (no accident) or obstructed (accident). We will show that the Poisson discrete distribution is an approximation assumed for modeling the crash process.

Other event detection methods are detailed in a variety of documents. For example, a survey of vision based automatic incident detection methods are found in [95]. [93] describes a method to improve safety at intersections using a combination of mathematically based algorithms and data mining. Yet another paper [94] describes the prediction of traffic accidents using a probabilistic model. In [96], a system was developed to monitor intersection traffic and use the tracking results to predict collisions over a short time period. A pixel-based strategy for the detection of unusual activities at intersections is described in [103]. Finally, [97] approaches vehicle collision avoidance from the perspective of the vehicle itself, which is equipped with sensors.

With more recent advances, some car manufacturers have added on-board systems, such as OnStar by General Motors [98], which will automatically notify a central monitoring station to report that the vehicle has just been involved in a crash.

Ohio State University scientists have created software that can identify traffic accident hot spots on state roadways, using the statistical information of injuries and fatalities from the highway patrol and other statistics about what makes accidents happen [99].

A Brief Overview of a Visual Surveillance System

Visual Surveillance is the process of monitoring an object's (people, vehicles, etc) activities in a scene. The usual approach requires capturing information from video recording devices, such as closed-circuit television cameras.

Visual surveillance systems address real-time observation of objects in some environments, resulting in a description about the behavior of the objects within that environment or among the objects themselves. It is utilized for measuring traffic flow, detecting accidents on highways, and routine maintenance [1, 2].

An intelligent visual surveillance system is the practice of monitoring and performing multiple surveillance tasks automatically by a computer vision system. It involves detecting and tracking objects, or blobs, in the video sequence and describing their actions. Using image analysis techniques, a visual surveillance system should be able to detect, identify and track objects, sense and segment motion, and record information captured by the surveillance cameras.

Much research has been done on traffic analysis [17-31], including vehicle identification, license plate detection, street crossing robots and autonomous vehicles. All of these solutions are context / domain specific. I have not yet discovered a complete "one size fits all" solution. With this in mind, the feature extractor and classifier can be made simpler, since we already know what we are seeking. There are many applications that are all specific to its own category or domain. Sometimes, it is not possible to determine the class membership of an object without examining the context in which the original object is embedded. There is no "global" solution to handle all computer vision problems / applications [10-11]. In this application, we expect to find vehicles and pedestrians in the street.

A Brief Overview of Computer Vision

Computer Vision is the essentially the science and technology of giving machines the ability to “see”. Computer Vision can be viewed as a combination of multi-disciplinary fields. Although earlier work exists, it was not until the late 1970s that an in-depth study of the field started as soon as computers could handle processing large data sets such as images. However, these studies typically began from various other fields, and as a result there is no standard approach for solving "the computer vision problem" [83]. Subsequently, there exists a large quantity of methods for solving a mixture of well-defined computer vision tasks, however, these methods frequently are task-specific and can rarely be generalized.

According to some researchers, computer vision technology is a branch of Artificial Intelligence that focuses on providing computers with the functions of typical human vision [16]. Others see Computer Vision as the study of understanding image content by computer or extracting information from visual stimuli.

One of the fundamentals of computer vision is enabling a machine / computer to interpret images in order to recognize and understand interesting objects in the picture, to complete a predefined goal or task.

The input to a computer vision system is usually a digital video stream, gray scale or color image, from a camera.

As said by Faugeras in [7], computer vision tries to answer the following questions

1. What information should be extracted from the visual sensors?
2. How is this information to be extracted?
3. How should this information be represented?

4. How must the information be used to perform the specified task?

The first three questions are mainly concerned with the low-level aspects of computer vision, such as edges, regions, blobs, etc. The fourth question deals with the high-level aspects of computer vision, which utilizes the low-level information obtained for surveillance applications or motion tracking applications.

Background

What is a digital image?

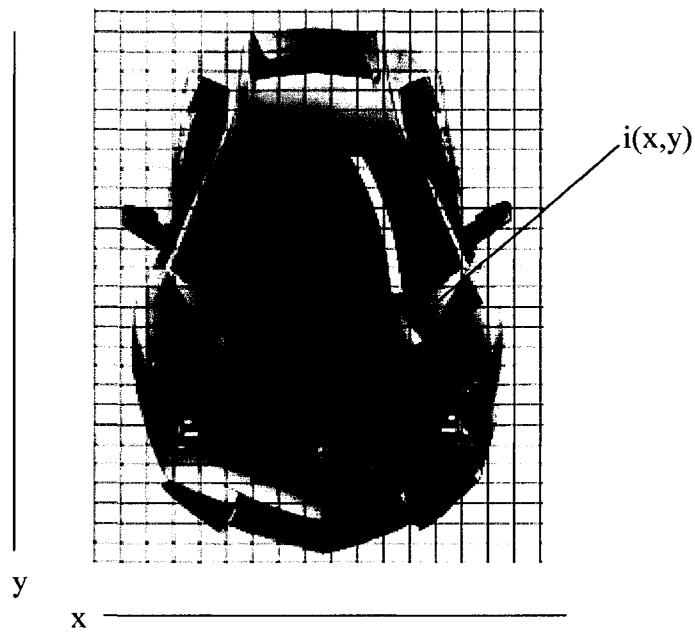


Figure 2 Pixel image array

For our purposes, an image is a two-dimensional array of monochromatic intensity values. The typical array sizes are 320x240, 512x512 or 640x480 (columns x rows), although lower and higher values are possible. Each array element is called a pixel (short for picture element). A typical video digitizer produces 8-bit pixels, with values in the range of 0 (black) to 255 (white). A pixel value represents light intensity (often called brightness or gray level). For color cameras, this information is usually separated into three image arrays, one for red, green, and blue.

Gray scale images contain less data than color images and are easier to process. However, the potential of color in helping with many aspects of computer vision processing is immense [83].

Color can be defined as the differences in the frequency of light. The wide range of colors that we see results when any of the three primary colors of light; red, green and blue, are combined together [45].

Color models can be defined in numerous ways, each with its own advantages and disadvantages. The RGB space is most commonly used with computer generated images. RGB is an additive color model.

The HSV (hue, saturation, value) color space, also known as HSB (where "B" represents brightness) varies the degrees of properties of colors to create new colors, instead of using a mixture of the colors themselves. HSV is a non-linear transformation of the RGB color space.

CIELAB or CIEXYZ color spaces are formal definitions of a color space, which were specifically designed to encompass all colors the average human eye can visualize. This is the most precise color space but is too complex for use.

What does recognition infer? Recognition is a word used to state the ability to identify and classify things based solely on previous stored information. What facilitates recognition? Most objects have explicit, distinctive, unambiguous features. These characteristic features can be visual, motion, shape, color, texture, or non-visual, taste, sound, or smell, in nature. However, in the case of computer vision, non-visual characteristics cannot be used.

Why is computer vision so difficult? The simple explanation is: there is too much information to process efficiently. Take for example, a typical monochrome video camera produces 320x240 pixels per image, thirty images per second, which is about two mega-bytes per second (Triple this value for color images – two mega-bytes for each color channel). For comparison, the human retina has more than one hundred million photo-receptors [32]. On the other hand, there is not enough information: the world is three-dimensional and dynamic, but (most) images are two-dimensional and static. How does vision produce vivid impressions of depth, shape, and motion from two-dimensional / flat images?

The ultimate goal of any computer vision system is to interpret the given visual data and to use that interpretation to complete a given task. A general scheme of a video surveillance system using computer vision can be found in [2].

Intelligent video surveillance in dynamic scenes has become an active research area [12]. However, many existing vision based systems are limited to measuring the traffic flow or queue detection [3, 4], or congestion detection on highways [5]. This survey details the current modern development of automated visual surveillance systems [6]. An intelligent control of traffic lights is detailed in [91].

Today, there is a widespread use of video recording devices in both private and public areas. Some examples include airports, banks, casinos, shopping malls, etc [8].

The goal of motion segmentation is to detect regions in the scene which correspond to moving objects like vehicles and pedestrians. It is one of the oldest computer vision problems and has been approached in many ways [9].

Feature recognition and / or object classification is the process for making decisions about the class membership of a pattern in question. Commonality among the scenes being portrayed is how class membership is reflected [33 - 36].

Common Problems

With any kind of visual image analysis, the following common problems occur:

- Cluttered Images - pictures which have a massive amount of content (clutter / noise) going on – or those pictures where there are multiple objects overlapping one another
- Foreground vs. Background - how do you tell what part of the image belongs to the background (uninteresting objects) and which part of the image belongs to the foreground (objects of interest)?
- Occlusion - when two objects overlap because one is in front of the other
- Real-time Performance - how to optimize your code, without giving up functionality. If your algorithm takes too long to classify an object, then its usefulness is diminished.
- Shadows - different lighting conditions and weather make this a common problem.
- Pose Variations- an object can look completely different if viewed from a different angle. This can be a problem if your algorithm matches based on

shape (template matching), orientation and position of the object with respect to the camera.

Image Acquisition and Pre-Processing

Image acquisition is the capturing of the data (images) to be analyzed. Pre-processing is the act of analyzing, processing and or transforming the data before it is passed on, to be used by the high-level segments of the system.

Histogram Equalization

Histogram equalization is a common method for improving the appearance of images. Suppose we start with an image that is predominantly dark. When its histogram is examined, it would be skewed towards the lower end of the grey scale with all the image elements being compressed into the dark end of the histogram. Stretching out the grey levels at the dark end to produce a more uniformly distributed histogram would then generate an image that is much clearer and brighter.

Histogram equalization can also be utilized when comparing a specific feature such as texture on two or more images. The process would first normalize the histograms to a standard histogram before the comparison was made. This is beneficial when the images have been acquired under different conditions, such as lighting or capture devices.

Noise Suppression

Image degradation is caused inadvertently during the image acquisition process. This can be caused by inappropriate illumination, motion, out of focus blurring, camera mechanical problems, noise or the quality of the digitized image being inferior.

There are several methods in which noise in an image can be reduced or eliminated. These include filtering, thresholding, convolution and mathematical morphology.

What is RGB color?

To understand color, you will need a brief overview of light. Visible light is made up of waves of varying colors, as shown below. The predominant colors of the spectrum are red, green and blue [14, 76].

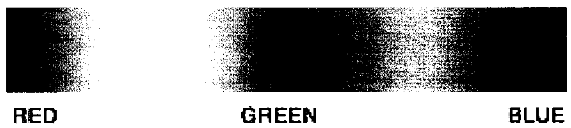


Figure 3 The color spectrum

Different levels of these colors can be combined to form other colors, since RGB is an additive color system.

Devices such as digital cameras, scanners and video cameras use the additive color system to gather information and reproduce the color image [14, 76].

With that in mind, an image is represented as an array of pixels. An RGB image subsequently is a compilation of the three planes that identify the amount of red, green and blue needed to categorize each pixel in the image and to simulate the appearance of color.

It can be time consuming to process color images since the three color planes (R, G, and B) must be processed together. To help resolve this issue and improve performance, images can be converted to a grayscale when necessary. Grayscale images are represented on one plane, where each pixel is assigned a value from 0 to 255 (for 8-bit images), where 0 is black and 255 is white, with all other values in between representing a shade of gray. To convert an existing RGB image to a grayscale image, the following formula

$$Y = (0.299 \times R) + (0.587 \times G) + (0.114 \times B)$$

Equation 2 Standard conversion of an RGB image to grayscale

can be applied to each pixel in the RGB image. Based on my research, there seems to be no one correct grayscale conversion, since it depends on the light sensitivity of the camera. The formula above seems to be the most common and works in the majority of cases.

Moving Object Detection (Motion Segmentation)

Once an image sequence is obtained and filtered, during the analysis of the objects in the images, it is necessary that we can distinguish between the objects of interest and the background. The techniques that are used to find these objects of interest are generally referred to as segmentation techniques. The concept of "segmentation" is generally known as finding the objects of interest in the image.

Background Subtraction

Background subtraction can be defined as detecting moving objects in a scene from a static camera. A commonly used method for detecting moving objects in a scene from static cameras is background subtraction [67]. Many diverse methods have been proposed with a variety of techniques. Generating a background image or model is a common principle in this approach. It is essentially detecting moving objects from the difference between the current frame and a reference frame, or collection of reference frames taken over time. The subtraction results in leaving only moving objects, which typically are processed at a later step. At the core, this background image must be a representation of the scene with no moving objects and must be frequently updated to adapt to the varying changes in the scene.

Developing a robust background subtraction algorithm has numerous challenges to overcome. One of the main challenges is that the background model must react timely to changes that occur in the scene, for example, starting and stopping of vehicles. The next challenge is that the algorithm must also adapt rapidly to changes in illumination. Another challenge is that the algorithm should not detect moving objects such as leaves, rain, snow or shadows. One of the most important problems is that it should perform well under real-time conditions. A good comparison and review of background subtraction techniques is detailed in [27] and [64].

Commonly, in an adaptive background subtraction method, a reference background frame is initialized at the start with the first few frames of the incoming video stream and then updated to adapt to dynamic scene changes over time. For each new frame, the foreground pixels are identified by subtracting the values from the reference background image frame and then filtering the absolute value of those differences with a dynamic threshold per pixel. The reference background and the threshold values are then updated by using the foreground pixel information. Some of the identified foreground pixels contain noise caused by errors, such as small movements from trees, etc. These isolated pixels can usually be filtered out by the use of morphological operations like erosion and dilation. Next, the individual pixels are grouped to form connected moving regions. These regions (which form the foreground mask) are displayed with a bounding box to be used in later processing steps.

Statistical Methods for Background Subtraction

More advanced methods have been developed to overcome the limitations of basic background subtraction methods. These methods use the characteristics of pixels to build a more advanced background model. Those statistics are also dynamically updated during processing. To identify which pixels belong to the foreground, each pixel's statistics are compared with that of the background model.

A good example is described in [65]. In their work, every pixel is separately modeled by a mixture of Gaussians which are updated online by incoming data. In order to detect whether a pixel belongs to the foreground or background, the Gaussian distributions of the mixture model for that pixel is evaluated.

The statistical model constructed in [71] represents each pixel as three distinct values: the minimum intensity value, the maximum intensity value, and the maximum intensity difference between consecutive frames examined throughout the training period. The parameters used for the model were updated periodically.

Temporal Differencing for Background Subtraction

Temporal differencing computes the pixel-wise difference between several successive frames in an image sequence in order to detect moving objects. While this method is adaptive to dynamic environments, it commonly fails to detect whole pertinent pixels of some of the moving objects.

[68] is an example of this method where detected moving targets in real video sequences using temporal differencing are described. Their method extracted moving sections by thresholding the motion difference image and then applying connected component analysis to cluster the motion into regions.

A better version of temporal differencing is introduced in VSAM [18] where they used three-frame differencing instead of two-frame differencing. They have successfully developed a hybrid algorithm for motion segmentation by combining an adaptive background subtraction algorithm with a 3-frame differencing technique. According to VSAM [18], this hybrid algorithm is very fast and more effective for detecting moving objects in image sequences.

Image Segmentation

Once the background subtraction algorithm has been applied, there will ultimately be some pixels in the resulting image that show up as moving, but should not be classified as valid moving objects. This can be due to tiny movements of the camera, image acquisition errors, image compression errors, etc. This is identified as noise. Even though the frequency of these pixels can be considerably reduced by accurate thresholding during the background subtraction stage, there will always be some noise generated. Therefore, a noise reduction algorithm is needed (noise can be reduced at both pre-processing and post processing).

The purpose of a noise reduction algorithm is to eliminate the random, unwanted pixels from the image that are not part of a valid moving object, but at the same time preserving the pixels that are part of valid moving objects.

The output image will inevitably contain small noise-generated blobs, once the background subtraction algorithm has been applied. So it is important that the resulting image be further refined. False blobs can be removed by the use of Gaussian filtering and simple area thresholding. The remaining blobs can then be cleaned using opening and closing morphological functions [77].

Thresholding

Gray level thresholding is a simple segmentation method. It is computationally inexpensive and fast. Thresholding creates a binary image from a gray-level one by changing all of the pixels below a certain threshold to 0 and all of the pixels above that threshold to 1 [90]. Formally, this can be represented as: If b(x,y) is the output from a(x,y) with some threshold T,

$$b(x,y) = \begin{cases} 1 \text{ if } a(x,y) >= T \\ 0 \text{ otherwise} \end{cases}$$

Equation 3 Formal definition of thresholding an image

Binary Images

Binary images are images that have been transformed into to two values, denoted by 0 and 1, but often with pixel values of 0 and 255, representing black and white respectively. Binary images are used in numerous applications since they are simple to process. Additionally, they can be useful when all of the information you need can be provided by the silhouette of the object, and when you can acquire the silhouette of that object easily.

Binary images are typically achieved by thresholding a color or a gray level image. Those pixels with a gray level above a certain threshold are set to 1 (or 255), while the rest are set to 0. However, choosing a threshold can be difficult [89]. Most approaches make use of the histogram, to show the number of times each gray level occurs in the image. Note that when the object and the background are relatively close in gray levels, it can be difficult to automatically determine the threshold.

We define the characteristic function of an object in a binary image to be:

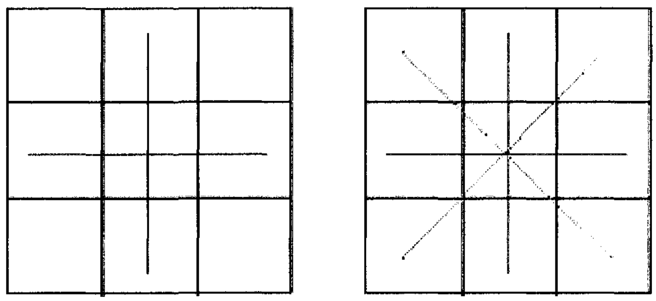
$$b(x, y) \begin{cases} = 1 \text{ for points on the object} \\ = 0 \text{ for the background point} \end{cases}$$

Equation 4 Formal definition of a binary image

Morphology

The root of mathematical morphology is the application of set operations to images. It helps mostly to segment images and emphasizes the role of shape. Normally, the image has already been thresholded to a binary image before the morphology operations are applied. The two most essential and fundamental operations in mathematical morphology are erosion and dilation. They are used to aid in the reconstruction of objects extracted from the background subtraction algorithm. The process of producing a new image from blending a function with an image is called convolution. Different masks can be used to obtain different images [83, 88].

In general, dilation produces objects that increase in size while erosion produces objects that reduce in size. The amount and the manner in which objects increase or decrease in size is completely dependent upon the choice of the structuring element. The 4-connected and 8-connected sets are the two most common structuring elements used.

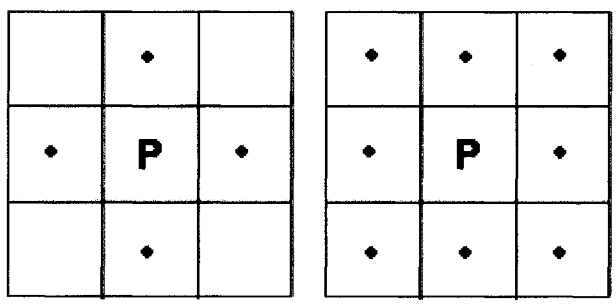


{
(-1, -1), (0, -1), (1, -1),
(-1, 0), (0, 0), (1, 0),
(-1, 1), (0, 1), (1, 1)
}

4-connected grid 8-connected grid Set of coordinate points

Figure 4 A 4 and 8 connected grid example

Two pixels, P and Q, are 4-adjacent if they are 4-neighbors to each other, and 8-adjacent if they are 8-neighbors of each other.



4-adjacent 8-adjacent

Figure 5 A 4 and 8 adjacent example

The outcome of dilation on a binary image is to gradually increase the boundaries of regions of pixels. As a result, areas of pixels increase in size while holes inside those regions decrease. The outcome of erosion on a binary image is to erode the boundaries of regions of pixels. As a result, areas of pixels decrease in size, and holes inside those areas become larger.

Opening and closing are two very important operations in mathematical morphology. They are both based on erosion and dilation. Erosion can be used to remove small clumps of unwanted foreground pixels quite effectively. However, it has the disadvantage of affecting all regions of the foreground pixels indiscriminately. Opening alleviates this by performing both erosion and dilation on the image. Dilation can be used to fill in small background holes in images. However, one of the challenges associated with this is that the dilation will also distort all regions of the pixels indiscriminately [83, 88]. 'Salt and Pepper' noise can be removed by using opening and then closing on the image.

The opening operation can split up objects or gaps that are connected in a binary image. The closing operation can fill in small holes. Both operations produce a certain amount of smoothing on an objects contour given a "smooth" structuring element. Opening smoothes from the inside of the objects' contour while closing smoothes from the outside of the objects' contour.

Formal definitions of the mathematical morphological operations:

$$\begin{array}{ll} \text{Erosion} & A \ominus B \\ \text{Dilation} & A \oplus B \\ \text{Opening} & A \circ B = (A \ominus B) \oplus B \\ \text{Closing} & A \bullet B = (A \oplus B) \ominus B \end{array}$$

Equation 5 Standard mathematical morphological operations where A is the image and B is the structuring element

The following figures are examples of the four most common mathematical morphological operations. The examples assume 8-connectedness.

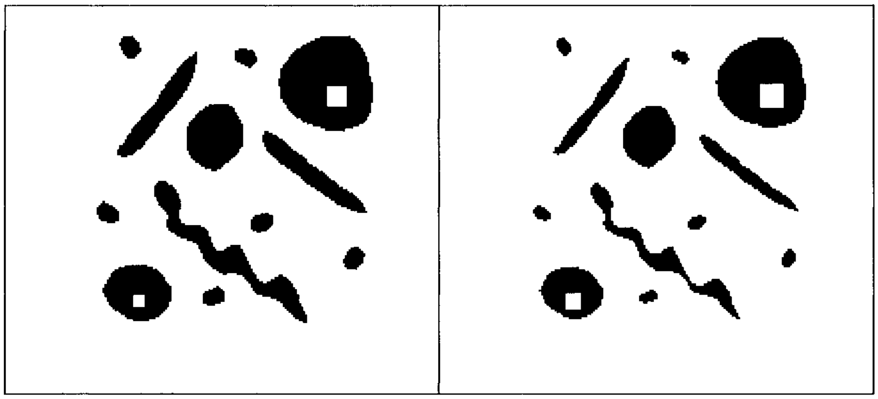


Figure 6 The effect of erosion using a 3x3 square structuring element

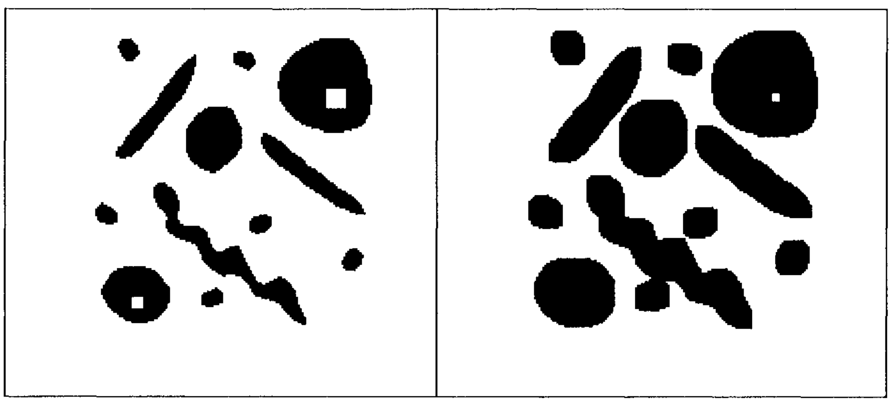


Figure 7 The effect of dilation using a 3x3 square structuring element

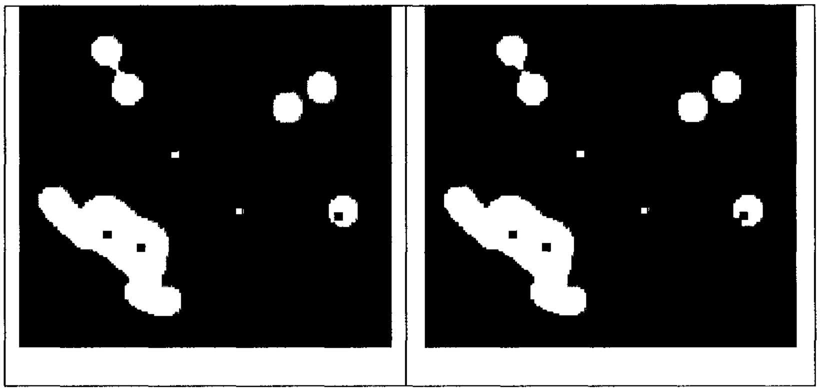


Figure 8 The effect of opening using a 3x3 square structuring element

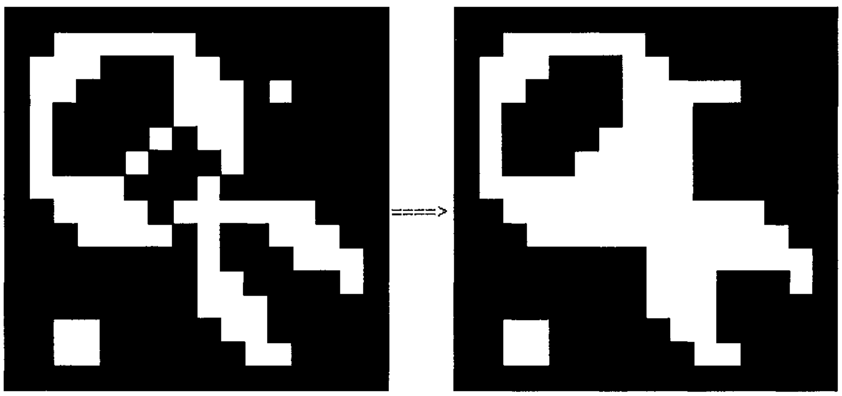


Figure 9 The effect of closing using a 3x3 square structuring element

In general, the operations here can help eliminate noise and irrelevant parts from images to obtain more accurate shape recognition.

Smoothing

Smoothing refers to changing the value of a pixel given the surrounding pixels. One way to smooth an image is to assign to each pixel the average of its neighbors. This will tend to cancel out extreme values.

Convolutions

A simple way of averaging nearby pixels is to use this convolution mask:

$$1/9 \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \text{ Blurring effect}$$

Figure 10 Blurring convolution mask

where the number in front of the mask weights all of the coefficients in the mask and is introduced to make sure that applying the convolution does not alter the mean intensity in the image.

Convolutions are the most general spatially invariant linear operators that can be applied to an image [83].

Gaussian Filter

A Gaussian filter, also known as a smoothing filter, is a convolution operator that blurs images while removing detail and noise. The output of a Gaussian filter is the weighted average of the pixels contained within the neighborhood of the filter mask. This filter acts as a low pass filter and preserves edges while eliminating noise. The Gaussian filter can be very useful when you need to detect edges and corners in an image. Applying a Gaussian filter will assure that the operator is rotationally invariant and will assure that the image noise is also reduced [51].

$$1/16 \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \begin{array}{l} \text{This mask} \\ \text{more closely fits} \\ \text{a Gaussian filter} \end{array}$$

Figure 11 Gaussian filter

Median Filter

The idea here is to find those pixels in the image which have extreme intensities and to ignore their actual values and replace them with more suitable values [83]. An apparent way of using this method is to apply a filter that prevents any pixel from having an intensity outside the range of its neighbors' intensities. This has the result of blurring the image and diminishing the edges in the image.

Color bleeding can occur if the three color channels, representing the red, green and blue channels in an RGB image, are processed independently. You can overcome this by using a vector median filter which processes all three color channels together.

Edge Detection

Edge detection is defined as finding those specific pixels that belong to only the borders of the objects. Edges can be considered as places in the image with a strong intensity contrast in only one direction, or where there is a sharp change in image brightness. Edge detection is only concerned with the image itself, and thus does not differentiate between different types of discontinuities in the image.

Edge detection provides a more meticulous means than thresholding for starting image segmentation. It is widely used and is an alternative path to image segmentation. It aids the image segmentation process by separating the image into areas corresponding to different objects, since edges normally happen at locations representing object boundaries. Edge detection also has a significant additional advantage in that it instantly reduces (by a large factor) the redundancy intrinsic in the image data.

There are many edge detection techniques available [39, 40, 52, 53, 74]. Some of the more common methods are described below.

Canny

The Canny Edge Detector algorithm has the following steps: [41, 42, 74]

1. Smoothes the image to eliminate the noise with a Gaussian filter
2. Computes the image gradient to highlight regions with high spatial derivatives
3. Applies non-maxima suppression to the gradient magnitude (moves along the regions and suppresses any pixel that is not at the maximum)
4. Reduces the gradient array by hysteresis. Hysteresis is used to track along the remaining pixels that have not been suppressed. Hysteresis uses two thresholds, if the magnitude is below the first threshold, it is set to zero (made a non-edge). If the magnitude is above the high threshold, it is made an edge. And if the magnitude is between the 2 thresholds, then it is set to zero unless there is a path from this pixel to a pixel with a gradient above T2.

Sobel

The Sobel Edge Detector uses two simple convolution kernels to create a series of gradient magnitudes; one to detect the changes in the vertical contrast (S_x) and the other is used to detect the horizontal contrast (S_y) [52, 74].

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad S_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Figure 12 Sobel convolution kernels

The outcome of each convolution is treated as a vector representing the edge through the current pixel. The pixel is marked as an edge if the magnitude of the sum of these two vectors is greater than a specific threshold.

There are a few disadvantages with the Sobel Edge detector. First, it is incredibly sensitive to noise in the source image. Second, resulting values from the calculation can easily overflow the maximum allowed pixel value [83].

Laplace

Laplacian based edge detection is based on the fact that the edge points of an image can be detected by finding the zero crossings of the second derivative of the image intensity [92, 74]. An edge has the one-dimensional shape of a slope and calculating the derivative of the image can highlight the location.

For example, an edge is shown by the jump in intensity: (Usually, edges will have higher pixel intensity values than those surrounding it)

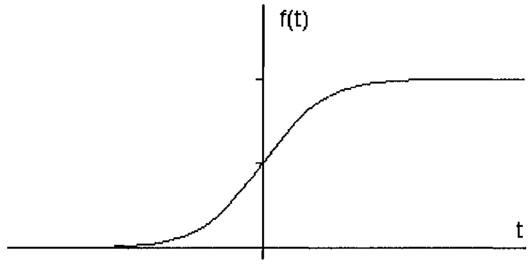


Figure 13 Sample edge

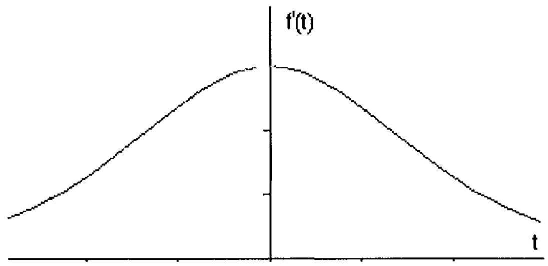


Figure 14 The first derivative, which is the gradient

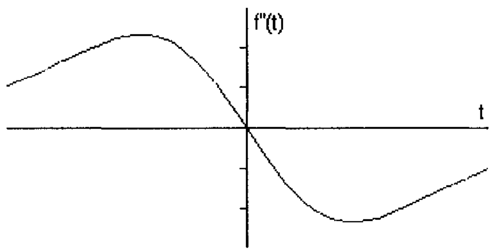


Figure 15 The second derivative

However, the calculations for the second derivative are very sensitive to noise. To rectify this, the Laplacian of Gaussian is normally used. This method combines Gaussian filtering with the Laplacian edge detection algorithm.

The Laplace operator can approximate the second derivative, but this only gives the gradient magnitude. The following convolution is used [74, 76]:

$$h = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \qquad h = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Figure 16 The Laplace operator

For a quick summary of edge detectors, please refer to Appendix B.

Edge detection has been an alternative path to image segmentation. It has a considerable additional advantage in that it immediately reduces the redundancy in the image data. This is useful because it drastically reduces both the space needed to store the information and the amount of processing later required to analyze it [83].

Connected Components (Object Labeling)

In general, binary images will normally contain more than just one object. We need to first identify the connected components in the image in order to identify or classify the objects found in that image. Connected components are the distinctly connected blobs that correspond to each object in the image [89].

Connected component analysis is a method used to find homogenous sections in a binary image. The groups identified by this step form a basis for object identification and tracking. Two pixels in a binary image are connected if and only if a path can be established between them. Commonly, 0's are used to identify the background of the image while 1's are used to identify the foreground of the image.

At this proposed step, the individual pixels are grouped and labeled to create connected moving regions. Additional processing groups disconnected areas together and removes fairly small sized regions. After grouping, each detected foreground object is represented by a bounding box. A common approach is to use a two-pass algorithm, in which you scan through and detect all connected regions, if two regions are touching (first pass), then go through again and re-label all touching regions with a single identifier (second pass).

The convex hull of the blob may be used to simplify complex shapes to provide a rapid hint of the scope of an object.

Moving Object Classification

The main goal of object classification is to precisely extract the region of interest corresponding to the type of object desired from all the moving regions detected by the motion segmentation step. These moving regions could relate to different objects in the real-world, such as vehicles, pedestrians, etc. There are several approaches toward moving object classification, shape and motion being the most relevant. Shape based methods rely on an objects' spatial information while motion based methods rely on an objects' temporal information for the classification.

Object detection methods are capable of searching for a specific class of object, like vehicle or pedestrian. In contrast, object recognition is the ability to identify specific instances of a class, such as the difference between my car and your car, not the difference between cars in general or what is not a car. These two types of techniques often go together, since the first step in recognition is to locate the object of interest.

Given that we are detecting objects in a video sequence, we have a much richer set of information available, specifically the dynamic information inherent in the video sequence.

For example, our feature selection for vehicle detection involves a class of vehicles, instead of a specific vehicle. We only need to detect the vehicle, no matter its make, model, color, etc.

Shape-Based Classification

Simple shape characteristics such as the geometric appearance, area, and silhouette of the detected objects are utilized in shape based classification methods in order to distinguish those objects from one another.

For instance, the method developed in [18] classified moving objects into four classes, such as vehicles, single human, human groups, and clutter, using a viewpoint-specific neural network classifier. The neural network used the area, separation, camera zoom magnification and the aspect ratio of the objects region as inputs.

In another instance, [68], used the dispersedness and area of the moving objects as classification metrics to help classify vehicles, humans, and other. Dispersedness is defined in terms of the object's area and contour length.

Another classification method proposed in [54] employs a logistic linear neural network with differential learning to distinguish two classes, people and vehicles.

Motion-Based Classification

Motion segmentation methods are based on motion cues, which are used to help distinguish the objects from the background. For example, these methods use only the temporal motion features of an object in order to classify its class [9, 55, 69]. They are commonly used to distinguish between rigid and non-rigid objects, such as vehicles from humans.

For instance, [9] described a self-similarity-based technique to detect and analyze periodic motion, by tracking the moving objects of interest. Their method shows that as an object that demonstrates periodic motion progresses, its self-similarity measure also shows periodic motion. The method makes use of this cue to classify moving objects using periodicity.

Optical flow analysis is also useful to distinguish rigid and non-rigid objects. The method described in [69] makes use of the local optical flow analysis of the detected object regions. In their algorithm, it was anticipated that rigid objects such as vehicles will depict little residual flow while non-rigid objects such as humans will depict a high average of residual flow. By using this cue, humans can be distinguished from other objects such as vehicles.

Color-Based Classification

Color provides more information for object recognition as compared to gray scale images. The idea here is that for color to be useful, it must bring the right sort of information to light on the task at hand [83]. In this instance, color can be used to aid in the segmentation of vehicles that are merged together in one large blob.

Object Recognition

Object recognition can be seen as the process of converting features of the image (such as edges, connected blobs, etc) into models of known objects (such as vehicle, person, etc). An objects appearance in a two-dimensional image is based upon its shape, pose, reflectance and lighting conditions.

With no reference available, how does a computer / machine "recognize" an object? It would have to start with its physical appearance, or more precisely, its features. It would have to be able to separate the foreground from the background and determine the object's dimensions relative to itself.

Humans tend to think of object recognition in terms of appearance. Therefore, pattern recognition features should reflect the nature of the objects that you are trying to classify. These features would include:

- Basic shape (objects characterized by its geometrical properties)
- Color
- Height, width, and depth
- Size or length
- Moving or stationary

Of course, a computer cannot make use of non-visual features, since its only input is from digital images. Therefore, those features cannot be included.

As we know, solid associations come from common connections. We tend to use a broad array of clues to recognize what is around us, even it we don't attempt to recognize it. Recognition itself consists of a set of clues, any one of which may be adequate enough to refine the identification process; however, it may be used to guide the computer to the correct conclusion. Motion also helps us define objects. Unique movement helps us to recognize particular objects. Furthermore, if any distinguishing features are missing, recognition becomes very difficult or impossible.

Recognizing three-dimensional objects from two-dimensional images is an important part of computer vision applications [37, 76]. While humans can recognize many diverse objects easily, visual recognition is generally a difficult task for a computer [38].

Shadow Detection and Suppression

Shadows can be defined as a region of darkness where light is blocked. All of the algorithms described above perform well on indoor and outdoor environments and have been used in real-time applications. Nevertheless, most of these algorithms are vulnerable to both local and global illumination changes. Shadows result in many motion detection methods failing to segment only the moving objects, given that the shadow moves with the object. If shadows are not eliminated during the motion detection phase, it could happen that two or more separate objects will be merged to together when the shadow is overlapping with another object.

Shadows can be grouped into two types, self-shadow and cast-shadow, according to [84]. An object's self-shadow is the part of the object which is not illuminated directly by the light source. In our case, the only relevant light sources are the sun, street lights and vehicle lights. An object's cast-shadow is the area projected onto the scene by the object. This area can be very large, particularly during the sunrise and sunset circumstances. A cast-shadow can be further divided into an umbra and penumbra [84]. When an object is fully opaque, that is all of the light is blocked directly by the object, the cast-shadow that occurs is identified as the umbra. When an object is partially transparent, that is only a certain amount of light is blocked, then the cast-shadow that occurs is identified as the penumbra. In this application, cast-shadows are the most important to eliminate. Self-shadows should not be removed since this could cause an incomplete object silhouette [72].

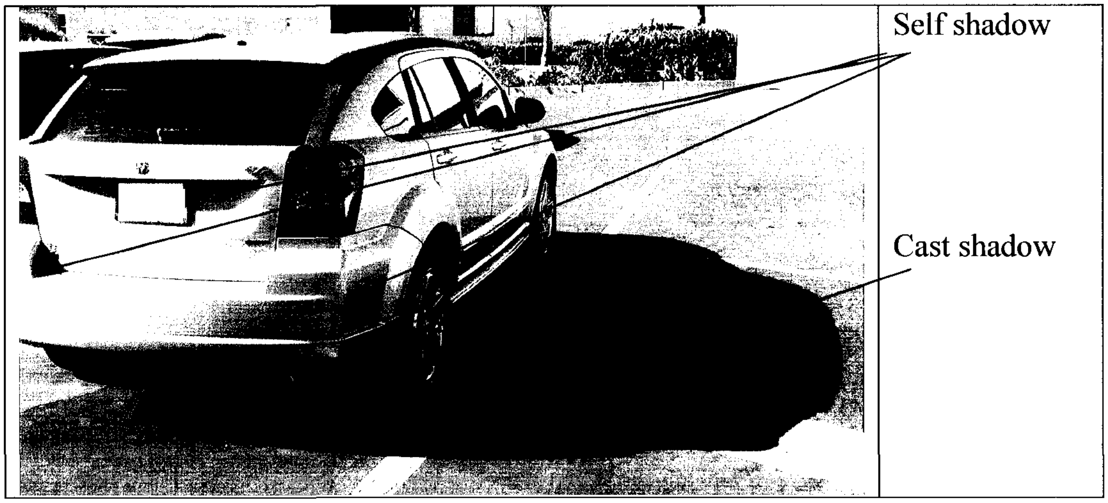


Figure 17 Shadow parts

Types of shadow elimination techniques, as stated in [73] are divided into two types, statistical approaches and deterministic approaches. Statistical approaches are simply a binary decision process where a pixel belongs to the foreground blob or its shadow [58, 62, 72]. Deterministic approaches add uncertainty by using probabilistic functions to express the class membership [56, 57, 59, 60, 61].

Statistical approaches are further broken down into parametric and non-parametric methods. In the parametric method, often two sources of data are used to help detect shadows and objects, local information which is represented by the individual pixel values and spatial information which is represented by as compact regions in the scene. A good example is the Anton system described in [63, 73]. In the non-parametric method, as described in [62], color is considered as a product of irradiance and reflectance. Brightness and chrominance distortion between the expected color of a pixel and its actual value is how the current image is computed. Shadows have similar chromaticity and lower brightness than the background [62, 100, 101].

Deterministic approaches are further broken down into model and non-model based methods. Model based approaches rely on the perception that the luminance ratio is a constant [60]. The luminance ratio is the ratio of the pixel intensity when it is under shadow compared to its appearance when under illumination. The reduction of a pixel's intensity in shadow regions is illustrated by a linear transformation. When a new image frame is processed, those pixels with the illumination reductions that follow the linear model are the marked as possible shadow pixels. The comparison between the background frame and the current frame is used by the non-model approach. It then sets a threshold to classify a pixel as shadow or non-shadow. For example is the Sakbot system as described in [73].

Finding Vehicles by Appearance

One approach could be to identify a certain pattern of pixels as the representation of a vehicle. Ideally, a lookup table would need to be created that contains every possible pixel pattern representation, attached to a binary value signifying the pattern's class as either "vehicle" or "non vehicle". The characterization would be a matter of retrieving the binary value at the pattern's location in the lookup table. Due to the memory restrictions of a computer, it would be difficult to store a lookup table with 400^{256} entries (20x20 pixel image fragment with 256 discrete levels), for the classifier [43].

A more realistic pattern classifier would need to search for simple features, which are specific to vehicles, since the above ideal pattern classifier is not feasible to implement.

Another approach might be to create silhouettes of the detected objects that could be extracted and used as the input for a template matching algorithm. Color histograms of detected objects could be stored and used to resolve identities during an occlusion (i.e. two vehicles crossing each other) [86, 78]. Different color spaces are invariant to different filters. For example, chromaticity spaces are invariant to intensity changes.

Tracking Moving Objects

Tracking moving objects can be characterized by observing the positions of an object or objects in a time sequence of images. The purpose of object tracking is to establish an association of objects or parts of an object between consecutive frames in the input video sequence. Tracking over time usually entails matching objects in consecutive frames using image features such as points, lines, position, velocity, shape, texture or color. An in-depth study of different tracking algorithms can be found at [47].

Object tracking can be divided into various categories according to different criteria according to the requirements of the application.

Optical Flow / Motion Estimation

Optical flow methods describe coherent motion between image frames over time to detect moving regions [85]. This segmentation is done by grouping motion vectors into groups having coherent motion to detect changes over time. This is a direct result of the relative motion of the viewer (in this case the camera) and the scene. Basically, given a set of points in an image, find those same points in another image. The resulting perceptible motion in the image is called the optical flow.

Most of the optical flow methods are computationally intensive, complex, and sensitive to noise and cannot be used effectively real-time without specialized hardware [70].

The Continuously Adaptive Mean Shift Algorithm (Camshift)

Camshift is based on the Mean Shift algorithm [107] for object tracking. It is mainly intended to perform efficient head and face tracking in a perceptual user interface [104, 105]. It is based on a variation of the Mean Shift algorithm that, given a probability density image, finds the mode of the distribution by iterating in the direction of maximum increase in probability density [106].

In order to use Camshift to track colored objects in a video scene, a probability distribution image of the desired color in the video scene must be created. Since vehicles can be any color, this method was not feasible in this application.

Kalman Filter

In computer vision applications that need to track objects, the Kalman filter can be used to model the behavior of each moving pixel, with predictions of its state and position in the next frame, given information about the pixel's initial state, velocity and acceleration. A Kalman filter is used to estimate the state of a linear system where the state is assumed to be a Gaussian distribution [46]. It works by estimating an unknown vector, based on an observed data vector, by finding the mean-squared estimate, where the estimate is chosen to minimize the error. The Kalman filter uses a recursive least squares algorithm to fit the data.

The use of this prediction of the object's position is valuable to the tracking processes. Once the objects are initially identified, the Kalman filter prediction would be able to tell where those objects are likely to be in the next frame.

The Kalman filter has been successfully used in the tracking of an active contour [48], the tracking of the boundary rectangle of a vehicle [49], and to track a two-dimensional silhouette of an object [50].

Occlusion

Occlusion can be described as the result of an object blocking another object from view. Occlusion can be categorized into three groups: occlusion by the background scene, self-occlusion or inter-object occlusion [47]. Background occlusion arises when a static structure in the background blocks the tracked objects. This normally happens when the tracked object moves behind it. Self-occlusion occurs when one part of the object blocks another. Inter-object occlusion occurs when two objects being tracked block each other. This situation happens most frequently when tracking moving objects. For example, when one object moves in front of or behind another object.

Color histograms can be useful for occlusion handling [86, 78]. The Kalman filter would also be useful in tracking objects and resolving occlusions [102]. However, the parameters used for the filter are difficult to estimate.

Challenges

There are many challenges to overcome in image segmentation, such as separating the foreground from the background, illumination changes (sudden or otherwise), noise, shadows, camouflage in color, etc [44].

A drawback of the statistical parametric approaches for shadow detection is that the parameters need to be selected very carefully.

Tracking can be a difficult task to apply to highly cluttered locations, due to inaccurate segmentation of the objects. Common segmentation problems are long shadows, occlusion of objects with each other or with static items in the scene. Dealing with these problems at the motion detection step is essential for robust tracking.

Evaluation

Classification errors should be minimized as much as possible. A classification error can be defined as making an error when it labels the unknown feature as class i, when its true class is j, and it is not part of the reject class.

CHAPTER 3

METHODOLOGY

Modeling Traffic Flow at a Signalized Intersection

A signalized intersection can be seen as the convergence of two or more roads, which is controlled by a traffic control device, commonly known as a traffic light.

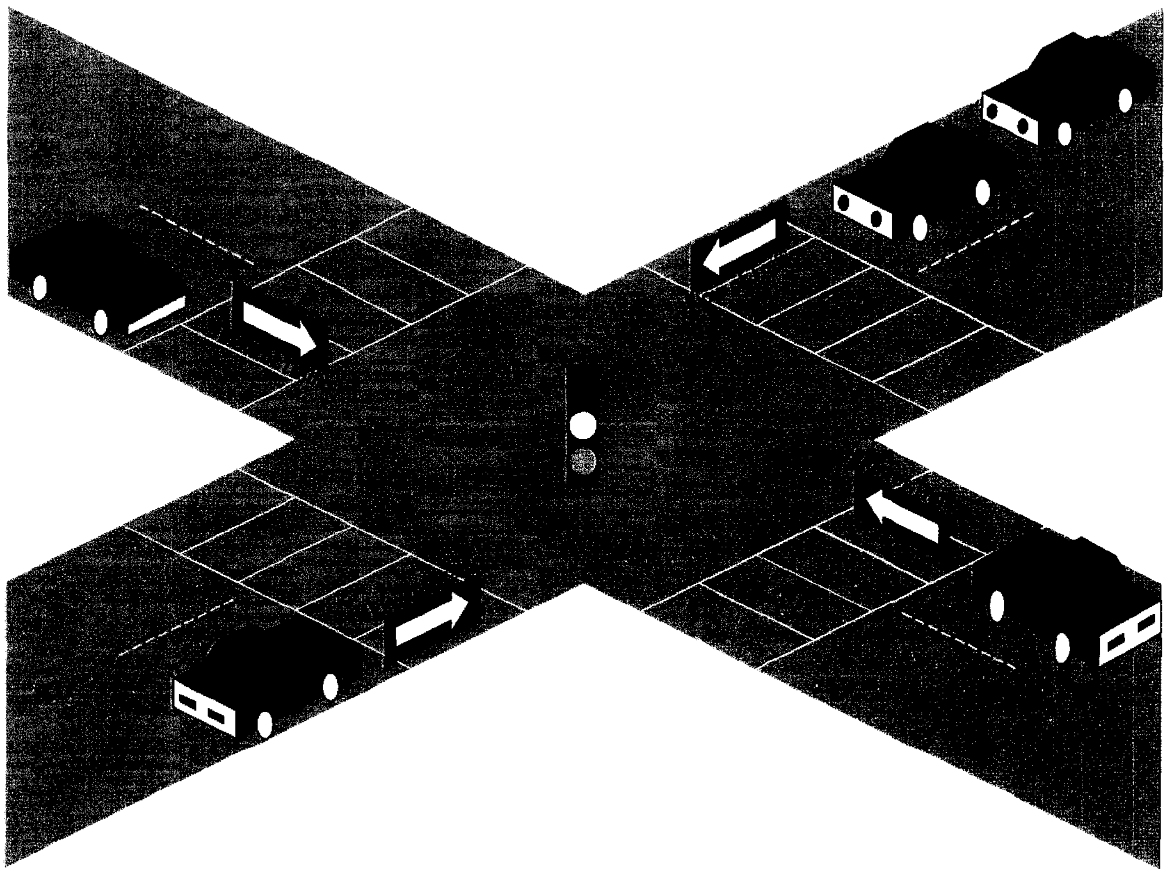


Figure 18 A typical intersection model

Traffic control signals are implemented for reducing or eliminating conflicts at intersections, and they also aid in controlling the flow of traffic. However, accidents still occur, whether it is from careless drivers, poor road or weather conditions, power outages, etc.

The model of intersection traffic is a somewhat simplified version of real-world intersection traffic, with all of the current traffic rules and regulations applied. Simulation models provide an alternative means for transportation studies when real models are not available.

As previously stated, we have modeled the road and the intersection as a service facility, which processes vehicles in a relatively small amount of time. A traffic accident is perceived as an interruption of that service.

With this in mind, we believe that the crash process can be modeled by using a two-state method, which implies that the intersection is in one of two states: clear (no accident) or obstructed (accident). We can then incorporate a rule-based system, which will help us in identifying that a crash has taken place.

In my analysis of signalized intersections, I discovered that there are certain areas in the intersection where the probability of an accident occurring is very high. I will label such areas as high-risk zones. These zones will help aid in the filtering of the data to help in the identification of accidents at intersections.

To assist the system with accident prediction, the following methods are proposed:

First, divide the intersection into four zones that indicate the beginning or ending of motion in and out of the intersection. Only two of the four zones would be 'active' at any time. The four zones could be used to set a flag if it detects that there is movement in one of the 'inactive' zones. For example, traffic is currently moving north – south (active zones). If any movement is detected in the zones that indicate east – west (inactive zones), then the probability of an accident is increased.

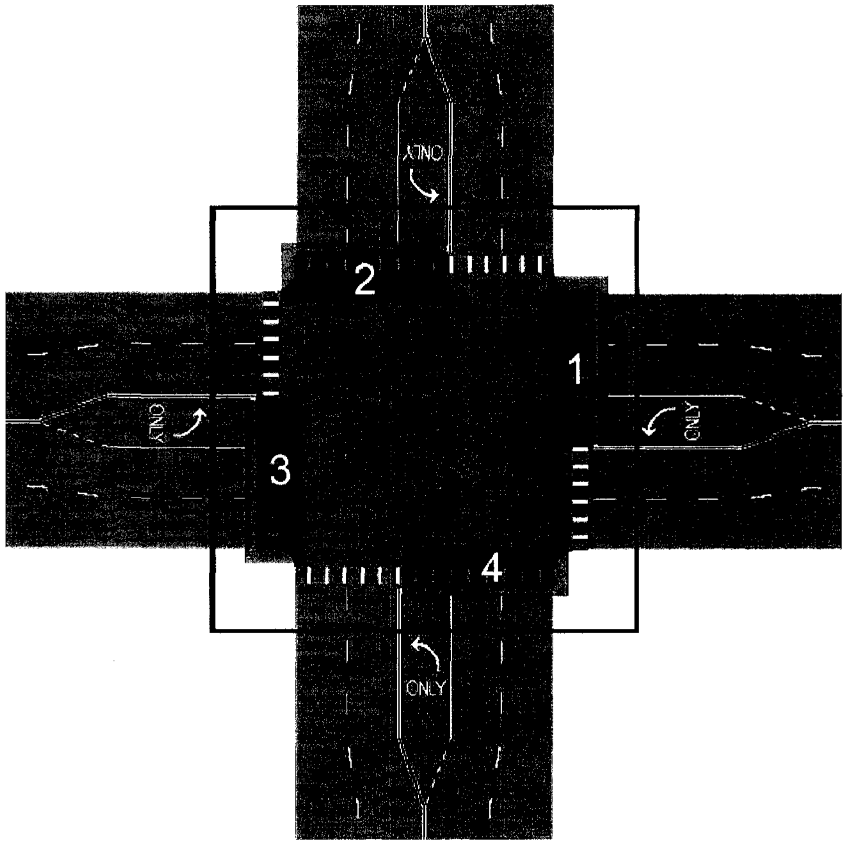


Figure 19 Active / Inactive zones

Second, the use of simple counters to count the number of vehicles making a left turn, a right turn or continuing their forward motion. Again, zones would need to be defined at certain areas in the intersection, close to the entrance (ingress) and exit (egress) points. The counter would be incremented at the ingress point and decremented at the egress point. A value of zero represents all traffic exited the intersection unobstructed. A positive value indicates that a vehicle entered the intersection but did not leave while a negative value indicates that a vehicle left the intersection from an unexpected point. The counters would be reset to zero in tandem with the traffic light timing.

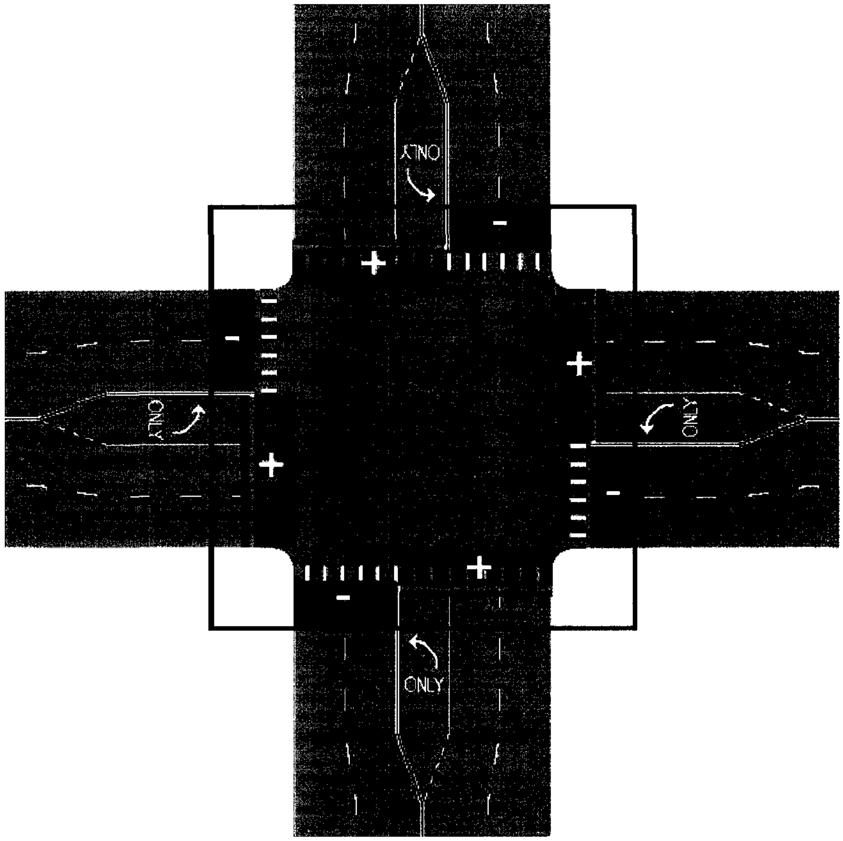


Figure 20 Ingress and Egress zones

Third, using motion vectors to track the flow of traffic. Traffic normally moves in a forward direction only in order to clear the intersection. In the case where momentum is detected in a backward or sideways motion, then the probability of an accident is very high. The motion is relative to the direction of the traffic flow.

Fourth, since tracking and predicting vehicular trajectories is an obstacle, using the 'most common' path would assist in overcoming the prediction challenge. Based on current traffic flow and common road rules, the intersection points of these paths would be the high-risk locations for an accident. This event would be triggered when opposing vehicles are detected crossing pre-defined thresholds at the same time.

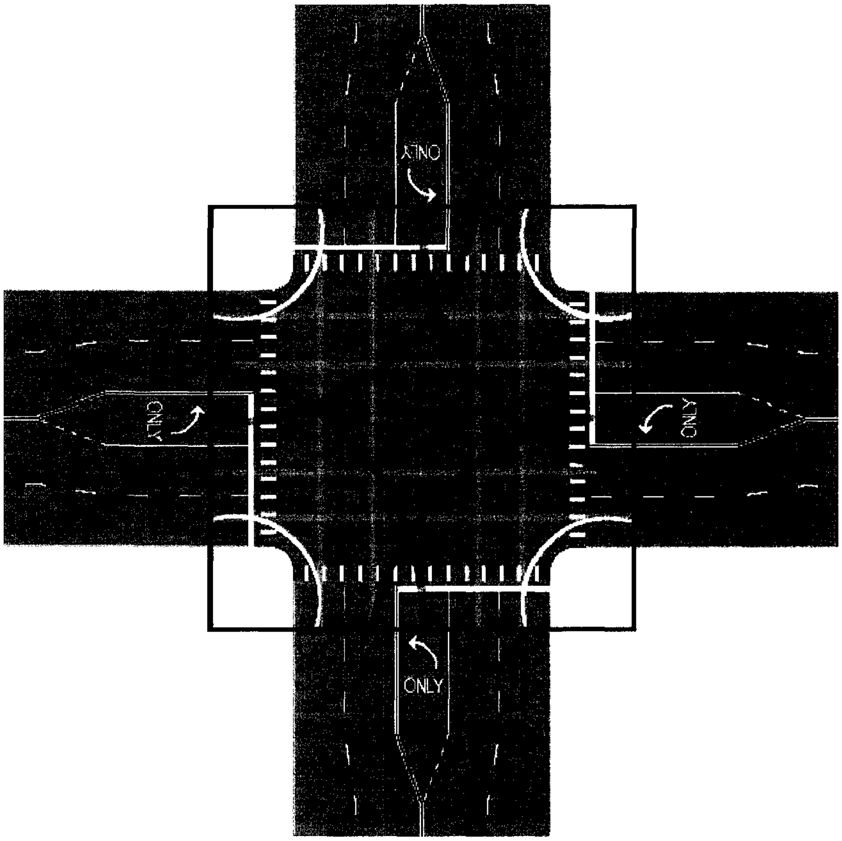


Figure 21 Common traffic paths

Fifth, with the following assumptions that vehicles follow specific geometric rules, and that the road surface is planar, we can use the lane markings in the road as natural dividers. Vehicles tend to stay parallel to each other governed by the lanes in the road. Vehicles also tend to be parallel with the center divider or center lane marking and the sidewalk. In the case where a vehicle is not parallel to the center line and/or sidewalk, would indicate a convergence and the possibility of an impending accident.

To assist the system with accident detection, the following methods are proposed:

First, once the system has identified a high probability of an accident occurring, as a final check, the system would compare the current image with a reference image of the intersection. If the difference between the current image and the reference image is significant, then an accident has more than likely occurred. The reference image is a static image of the background, in this case, the intersection clear of traffic, which is updated at periodic time intervals (every second or every minute), in order to stay current. The time interval would need to be adjusted if too many false positives are occurring because the reference background image is too far out of date. This would include illumination changes representative of the current time of day and prevailing weather conditions.

Second, the system could keep a history of all tracked objects in the intersection. The object would be added to the history once it entered the intersection and subsequently removed from the history when it leaves the intersection. If the tracked object stops moving, but has not left the intersection, it would still be in the history. This could indicate the presence of an accident.

One problem related to these scenarios occurs when the intersection is 'grid locked', i.e. the intersection is blocked with traffic and it is not moving. This would result in a false positive, since some of the above scenarios would be satisfied.

Once the system has identified the probability of an accident, this information would need to be sent to a central monitoring location for human verification. If an accident has occurred, the monitoring facility could 'communicate' with the intersection and then notify emergency response vehicles, based on the severity of the accident.

While my approach focuses on the use of only one camera, there are two other scenarios that could be implemented: using two cameras on opposite corners of the intersection or four cameras, one on each corner of the intersection. The same processes would be utilized for all cameras.

Review of the Crash Statistics

My proposed approach starts with a mathematical model that pertains to the number of accidents that have occurred in an intersection.

Table 1 Statewide crash statistics for the state of Nevada over the last 5 years [80]

Year	Total Crashes	Total Crashes reported by Nevada Highway Patrol	Total Crashes at Intersections	Percentage (Intersection / Crashes)	Percentage (Intersection / Total crashes – Total from Highway Patrol)
2002	62,237	15,973	23,694	38.07	51.21
2003	63,582	15,937	22,815	35.88	47.89
2004	59,657	15,311[1]	18,968	31.80	42.77
2005	61,487	15,781[1]	17,314	28.16	37.88
2006	61,142	15,692[1]	16,511	27.00	36.33
Total	308,105	78,693	99,302	32.18	43.22

The above data was extracted from two data sources. In late 2003 through 2004, the state was transitioning to a new data collection system with completely different data elements. Through 2006, not all crashes were in both systems [80].

By taking the total number of crashes, and subtracting the total number of crashes reported by the highway patrol, this gives us the total number of crashes that occurred on

[1] Actual audited data is not available as of February 2008. These represent approximately 26% of the total number of reported crashes. There is no facility to approximate the number of accidents that have gone unreported, and this number could be substantial.

surface streets (non-highway). In using this number, we have a better idea of how many crashes on non-highway roads were at intersections.

The mathematical model shown is a Poisson distribution that predicts the number of accidents in any intersection per week, which can be used as approximations for modeling the crash process.

The Poisson distribution model was used as a basis for the computation of the probability of accidents in an intersection. The raw crash data was used to extrapolate those intersections with the highest probability of incidents, and to aid in the prediction of accidents.

Recorded data for the following intersections, on a statistical basis, fits the Poisson distribution model. The raw crash data was split into groups that were representative of a week, with 52 weeks in a year. The raw crash data spanned a 5-year period, beginning January 2002 and ending in December 2006.

The following formula was used for the calculation:

$$P(X = k) = \lambda^k e^{-\lambda} / k! \text{ where } k = 0, 1, 2\ldots$$

Equation 6 Function used to calculate the Poisson distribution

where X is the random variable that signifies the number of accidents per the unit of time. The unit of time used for the calculation of the Poisson distribution is 1 week. λ indicates the average number of accidents per the unit of time. $1 / \lambda$ signifies the time between accidents. The inter-arrival time is an exponential distribution of the number of accidents in the intersection:

$$f(x) = e^{-\mu x} \text{ where } \mu = 1 / \lambda.$$

Equation 7 Inter-arrival time

The variance of the number of accidents should be close to λ, that is why this is a Poisson distribution.

For example, the following information was collected for the intersection of Ann and Simmons for the year 2004.

Table 2 Ann and Simmons weekly total of accidents for 2004

Week #	# of accidents		Week # (con't)	# of accidents (cont)		Week # (con't)	# of accidents (cont)
1	0		19	0		36	0
2	0		20	0		37	0
3	0		21	0		38	0
4	0		22	0		39	0
5	1		23	0		40	0
6	1		24	0		41	0
7	0		25	0		42	0
8	2		26	0		43	0
9	0		27	0		44	0
10	0		28	0		45	0
11	0		29	0		46	0
12	1		30	1		47	0
13	0		31	0		48	0
14	1		32	0		49	0
15	0		33	0		50	1
16	1		34	0		51	0
17	1		35	1		52	0
18	1						

Table 3 The maximum number of accidents per week was 2 for Ann and Simmons

Observed	Count	Count / 52	
count = 0	41	0.7885	The # of weeks with 0 accidents
count = 1	10	0.1923	The # of weeks with 1 accident
count = 2	1	0.0192	The # of weeks with 2 accidents
Total	*52*	*1.0*	

Table 4 There were a total of 44 weeks without an accident for Ann and Simmons

Mean	Variance	Lambda (mean+var/2)	Count = 0	Count > 0
0.2308	0.2202	0.2255	41	11

There are 11 weeks where at least one accident occurred leaving 41 weeks without an accident. The mean for all 52 weeks is 0.2308. That is the sum of all accidents / number of weeks in a year which is 12 / 52. The variance for all 52 week is 0.2202. The variance is a common measure of describing the spread of observations in a distribution. λ is the average of the mean and the variance: 0.2308 + 0.2202 / 2 = 0.2255.

To manually calculate the variance:

1 week has (2 – 0.2308) = 1.7692 (# of accidents minus the mean)

10 weeks have (1 – 0.2308) = 0.7692

41 weeks have (0 – 0.2308) = -0.2308

$(1 * 1.7692^2) + (10 * 0.7692^2) + (41 * -0.2308^2) / (52 - 1) = 3.1301 + 5.9167 + 2.1840 / 51 = 11.2308 / 51 = 0.2202$

Table 5 Poisson distribution for Ann and Simmons

Theoretical	Poisson		
X	P(X) /Lambda	P(X) /Mean	P(X) * 52
0	0.7981	0.7939	41.5012
1	0.1800	0.1832	9.36
2	0.0203	0.0211	1.0556
3	0.0015	0.0016	0.078
4	0.0001	0.0001	0.0052
			52.00

P(X) is always a number between zero and one with one being the highest probability

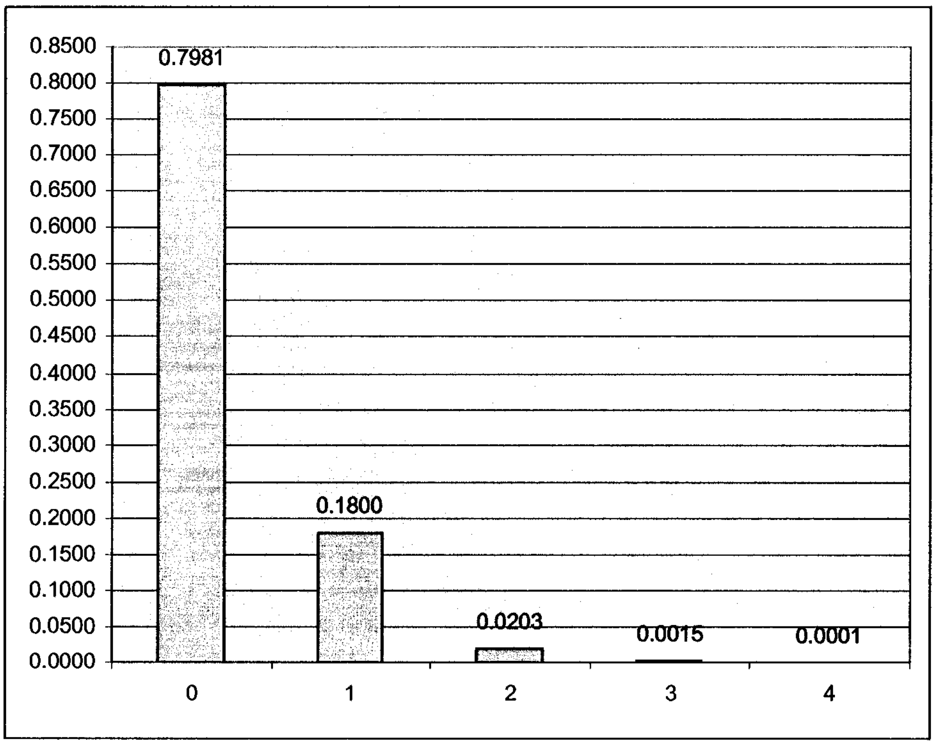


Figure 22 Poisson chart for Ann and Simmons

For other statistical information on the twelve other intersections gathered, please see Appendix A.

Implementation and Specifics

There are a number of different approaches to vehicle classification and tracking. The majority of tracking systems identify objects by their motion. The extent to which the vehicle classification and tracking depends on the background and lighting conditions at the time.

With so many diverse methods to choose from, it is hard to choose the right one for the task at hand. Part of my code development and research will be to see which method works the best, for the current problem that I am trying to solve.

The Intel Open Source Computer Vision Library [13] was used in the experimental system. It is a collection of C function and C++ classes that implement many algorithms of image processing and computer vision.

Moving Object Detection (Motion Segmentation)

Adaptive Background Subtraction Model

In my experimental project, moving objects are detected and handled by use of an adaptive background subtraction method, like those described in [66], which works well for outdoor environments.

The function RunningAvg calculates the weighted sum of two images. Once a statistical model is available, slow updating of the value is often required to account for slowly changing lighting, etc. This can be done by using a simple adaptive filter:

$$\mu_t = \alpha y + (1 - \alpha)\mu_{t-1},$$

Equation 8 Adaptive background subtraction function where μ is the updated value, $0 \leq \alpha \leq 1$ is an averaging constant, typically set to a small value such as 0.05, and y is a new observation at time t. When the function is applied to a frame sequence, the result is called the running average of the sequence. The value of 0.015 was used in this project for α (alpha).

A running average was selected, since the application cannot assume that no moving objects will be present in the scene when the system is started. Therefore, the background model is updated frequently, and a good background model is generated.

Noise Suppression

A smoothing filter based on Gaussian was used on the color image before it was thresholded to binary. A 5 x 5 matrix was used. This appeared to give the best results.

Thresholding, Erode, Dilation

These operations will be used after the background subtraction step to:

- Remove objects too small to be valid trackable regions
- Connect moving objects that have been split by a noisy background
- Fill in the holes within moving objects that may appear due to the background subtraction technique or noise
- "Normalize" the image for further processing

One very special property of erosion is its ability to find the contour of an object by subtracting the original image and the result of the erosion. However, this method does not replace edge detection.

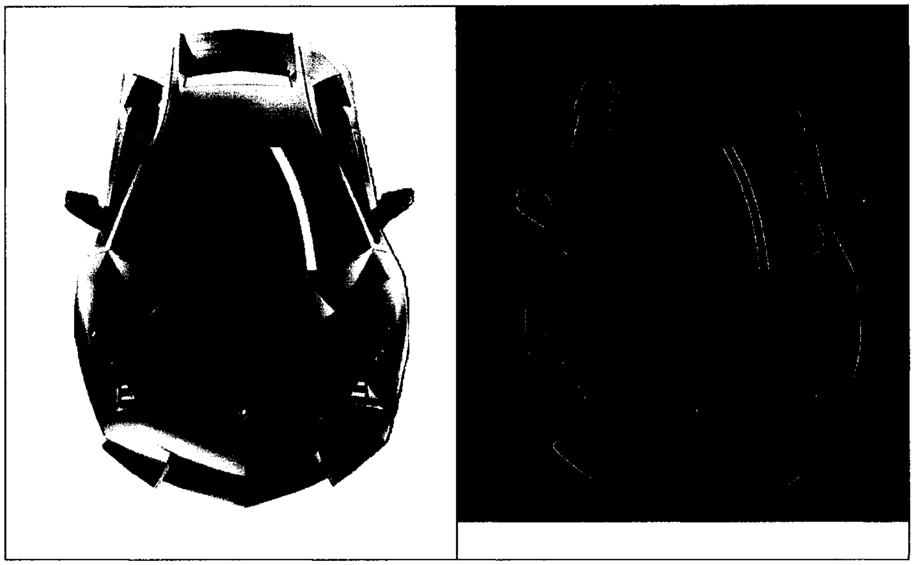


Figure 23 Example of contour by erosion

Edge Detection

I will be implementing the Canny Edge Detection algorithm, since it is an optimal edge detector [41, 42], and works well in most cases.

Shadow Detection and Suppression

Shadows pose a problem to any surveillance system, since the shadows move with the object. To assist with shadow detection and removal, the following methods are proposed:

First, the time of day and time of year determine the location of a shadow, relative to the object that it is being projected from. This information would be utilized in order to assist the shadow removal process.

Second, using a histogram would help in determining which pixels were shadow pixels. A histogram with a majority of low values would indicate a dark region in the image. This information would be utilized to assist in the shadow removal process.

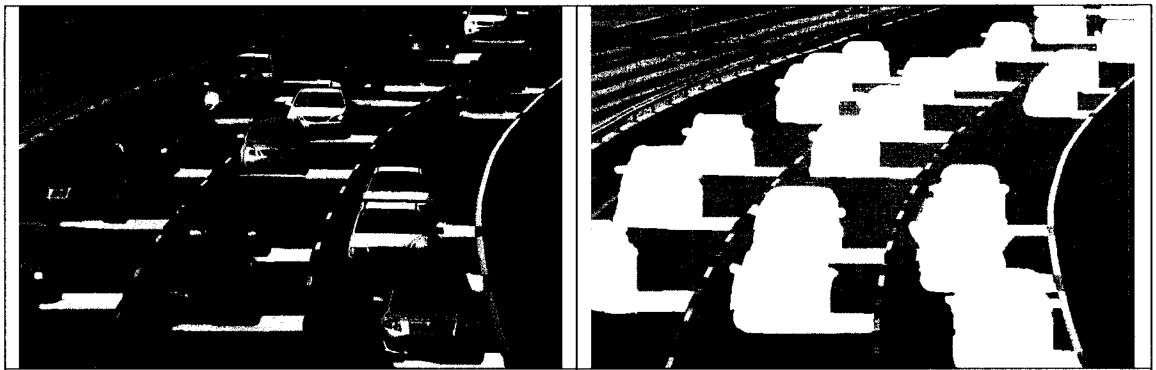
Values less than 50 would be marked as possible shadow pixels. A potential problem with this is that dark colored vehicles or those with dark tinted windows could create a false positive.

Third, shadow regions could also be 'trimmed' when they are too long and cross lane boundaries. We can trim the detected blobs region at the lane divider.

Figure 24 Original Image

Shadows are highlighted	Vehicle segmentation with shadows

Figure 25 Image with detected shadows before truncation



Shadows are highlighted	Vehicle segmentation with shadows trimmed at lane boundary

Figure 26 Image with shadows truncated

An accident implies that no objects are moving, once it has occurred. Since this system is built around moving objects, shadows do not hinder the accident detection process in a significant way.

Object Classification

In order for the classification to take place, features must be extracted that are common to the objects in question. Height and width are common attributes that can be calculated and might make good features. My goal is to find simple, specific, vehicle features which are invariant between different vehicle types, reflectivity, orientation and lighting.

Since this system is going to be used to examine traffic scenes, the two main object classes are vehicle and pedestrian. Other object sub-classes can be based on these two main ones, like car, truck, and bicycle. Objects that should be rejected from the classifications are trees, mailboxes, buildings, street signs and other non-moving objects.

Searching for objects by their features is far more efficient than template matching. To assist in the classification of vehicles and pedestrians, the following methods are proposed:

First, we know that vehicles have certain physical properties, such as proportional height and width and they must fit into the lane. Vehicles also tend to be 'boxy' or rectangular in their appearance. Humans are taller and narrower than vehicles, while vehicles are shorter and wider and longer than humans.

Second, we can use the lane markings to note the width of the vehicle. We can also determine the length of the vehicle as it passes by the camera.

To find a perpendicular line from any two lines: start with a point on one of the lines then sweep across the other line. The point chosen on the first line should be near the center of the area that will be examined. Whichever line is the shortest is the one that is perpendicular (and the straightest line).

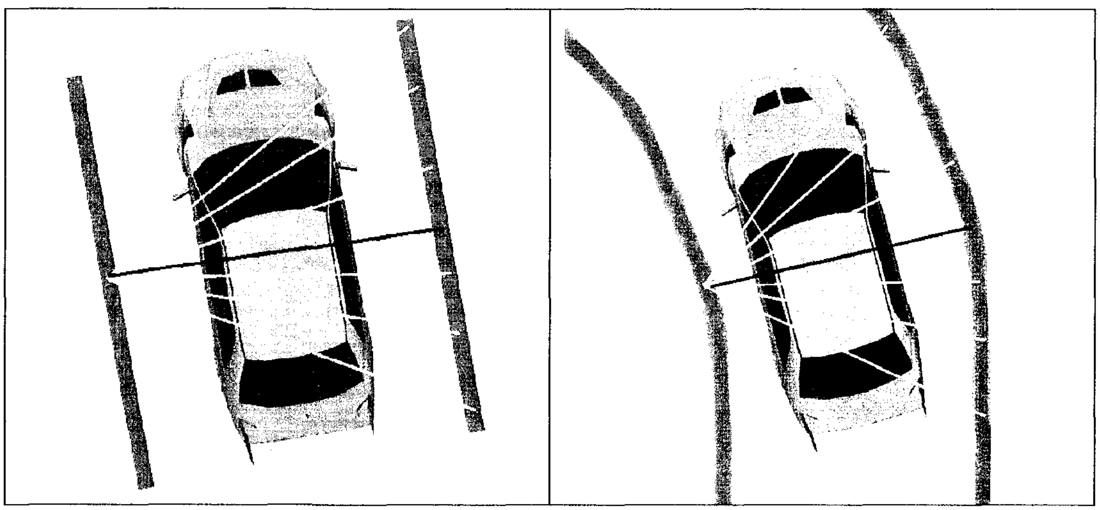


Figure 27 Illustration of finding the shortest line

Tracking Moving Objects

Only the objects that move in the ROI (region of interest) are of significance. The intersection itself is the ROI.

Kalman filtering was not implemented to track moving objects due to its complexity and it is hard to estimate the parameters needed for accurate tracking and prediction.

Occlusion

Occlusion occurs as vehicles pass each other in the intersection, from the camera's perspective. The objective of this thesis is to detect an accident. An accident implies that objects that were moving are no longer moving. There are times when two or more vehicles are detected as one due to improper segmentation due to shadows or vehicles moving past each other, etc. However, it was not necessary to utilize occlusion handling in this application as it is based on motion. As long as vehicles are in motion, occlusions are irrelevant.

CHAPTER 4

RESULTS AND CONCLUSIONS

As previously stated, due to an NRS regulation, all data was simulated, given that real data was not available. Also, since accidents are rare events, they are not often caught on tape.

The use of the statistical models aids us in identifying intersections that have a high probability for the potential for an accident to occur.

The use of computer vision techniques would be beneficial in the automatic detection of accidents at intersections. It could decrease the amount of time that an intersection was obstructed since emergency vehicles would be notified almost as soon as the accident happened. The monitoring location could determine the severity of the accident and notify police to assist in the redirection of traffic away from the scene. The potential to save lives is increased due to the quicker arrival of an ambulance and EMT personnel.

This technology could be further utilized if the digital cameras were equipped with night vision capabilities. Color is the only loss of information, since night vision would be inherently black and white.

Future Work

It would be beneficial for the system to automatically segment the road based on the lane markers or other cues in the road. Historically, the Hough Transform has been the main method of detecting straight lines.

This process could be further extended to control the traffic flow in the intersection. It would be simple to detect any moving objects in the intersection when the traffic light is changing from green to red. If there is movement, the traffic light could be delayed before turning green until the intersection is clear of traffic. This could prevent vehicles from running the red light and blocking the current right of way. My idea addresses the problem of traffic analysis from a different perspective that of the traffic signals, since the traffic signals control the traffic flow. This is my presentation of Smart Traffic Lights at intersections for the prevention of accidents and to assist in traffic flow, control and analysis.

APPENDIX A

DATA ANALYSIS

The intersection of Las Vegas Blvd and Flamingo Road for 2005

Table 6 Total number of accidents by week for one year for Las Vegas Blvd and Flamingo Road.

Week #	# of accidents		Week # (con't)	# of accidents (con't)		Week # (con't)	# of accidents (con't)
1	0		19	3		36	2
2	4		20	3		37	3
3	1		21	3		38	5
4	2		22	2		39	2
5	3		23	2		40	2
6	0		24	0		41	5
7	2		25	7		42	1
8	3		26	2		43	4
9	3		27	1		44	3
10	2		28	2		45	3
11	1		29	4		46	0
12	1		30	3		47	2
13	2		31	1		48	6
14	2		32	3		49	0
15	4		33	0		50	0
16	1		34	3		51	2
17	3		35	3		52	4
18	2						

Table 7 There were a total of 7 weeks without an accident for Las Vegas Blvd and Flamingo Road

Mean	Variance	Lambda (mean+var/2)	Count = 0	Count > 0

2.3462	2.3876	2.3669	7	45

Table 8 The maximum number of accidents reported in one week was 7 for Las Vegas Blvd and Flamingo Road

Observed	Count	Count / 52	
count = 0	7	0.1346	*The # of weeks with 0 accidents*
count = 1	7	0.1346	*The # of weeks with 1 accident*
count = 2	15	0.2885	*The # of weeks with 2 accidents*
count = 3	14	0.2692	*The # of weeks with 3 accidents*
count = 4	5	0.0962	*The # of weeks with 4 accidents*
count = 5	2	0.0385	*The # of weeks with 5 accidents*
count = 6	1	0.0192	*The # of weeks with 6 accidents*
count = 7	1	0.0192	*The # of weeks with 7 accidents*
Total	*52*	*1.0000*	

Table 9 Poisson distribution for Las Vegas Blvd and Flamingo Road

Theoretical	Poisson		
X	P(X) /Lambda	P(X) /Mean	P(X) * 52
0	0.0938	0.0957	4.8761
1	0.2219	0.2246	11.5413
2	0.2627	0.2635	13.6585
3	0.2072	0.2061	10.7760
4	0.1226	0.1209	6.3764
5	0.0580	0.0567	3.0185
6	0.0229	0.0222	1.1907
7	0.0077	0.0074	0.4026
8	0.0023	0.0022	0.1191
9	0.0006	0.0006	0.0313
10	0.0001[2]	0.0001	0.0074
			52.00

[2] This value does not show on the chart – it is de minimis.

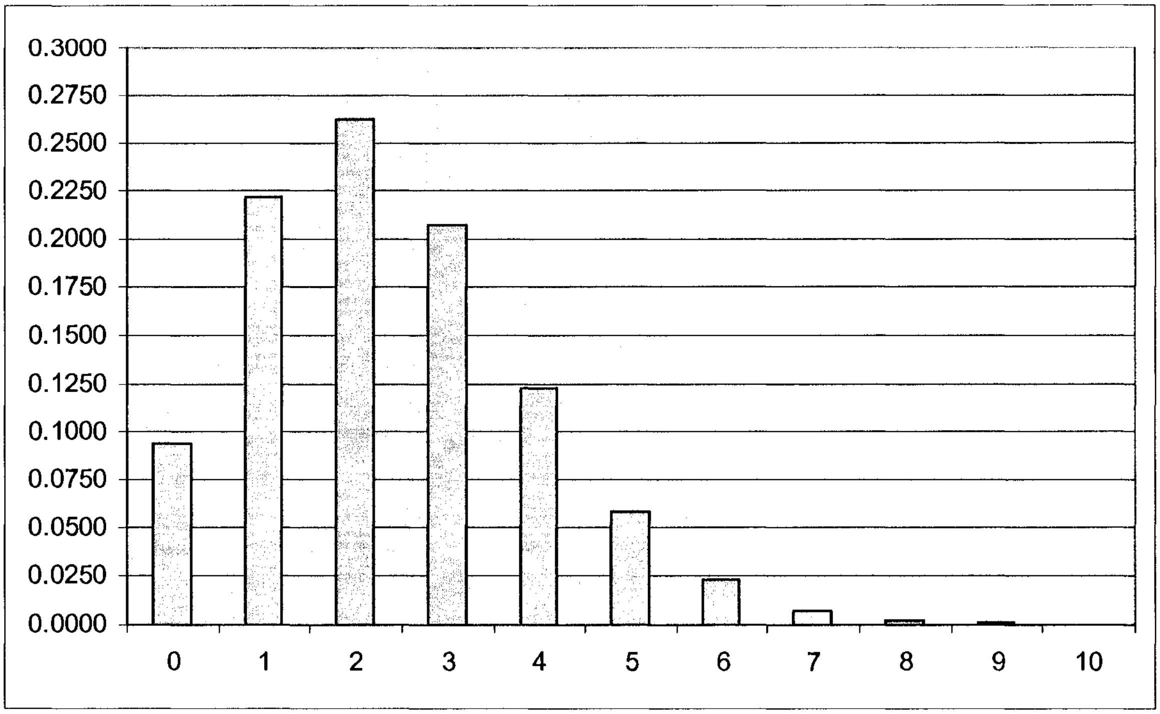


Figure 28 Poisson chart for Las Vegas Blvd and Flamingo Road

The intersection of Main and Charleston for 2005

Table 10 Total number of accidents by week for one year for Main and Charleston

Week #	# of accidents		Week # (con't)	# of accidents (con't)		Week # (con't)	# of accidents (con't)
1	0		19	1		36	1
2	0		20	0		37	0
3	1		21	0		38	3
4	2		22	2		39	0
5	0		23	0		40	2
6	1		24	0		41	0
7	1		25	1		42	0
8	1		26	1		43	0
9	2		27	2		44	2
10	0		28	1		45	0
11	0		29	1		46	0
12	1		30	0		47	0
13	1		31	0		48	0
14	1		32	1		49	0
15	0		33	0		50	0
16	1		34	0		51	0
17	0		35	2		52	1
18	0						

Table 11 There were a total of 28 weeks without an accident for Main & Charleston

Mean	Variance	Lambda (mean+var/2)	Count = 0	Count > 0
0.6346	0.6286	0.6316	28	24

Table 12 The maximum number of accidents reported in one week was 3 for Main and Charleston

Observed	Count	Count / 52	
count = 0	28	0.5385	*The # of weeks with 0 accidents*
count = 1	16	0.3077	*The # of weeks with 1 accident*
count = 2	7	0.1346	*The # of weeks with 2 accidents*
count = 3	1	0.0192	*The # of weeks with 3 accidents*
Total	*52*	*1.0000*	

Table 13 Poisson distribution for Main and Charleston

Theoretical	Poisson		
X	P(X) /Lambda	P(X) /Mean	P(X) * 52
0	0.5317	0.5301	27.6505
1	0.3358	0.3364	17.4640
2	0.1061	0.1068	5.5151
3	0.0223	0.0226	1.1611
4	0.0035	0.0036	0.1833
5	0.0004[3]	0.0005	0.0232
			52.00

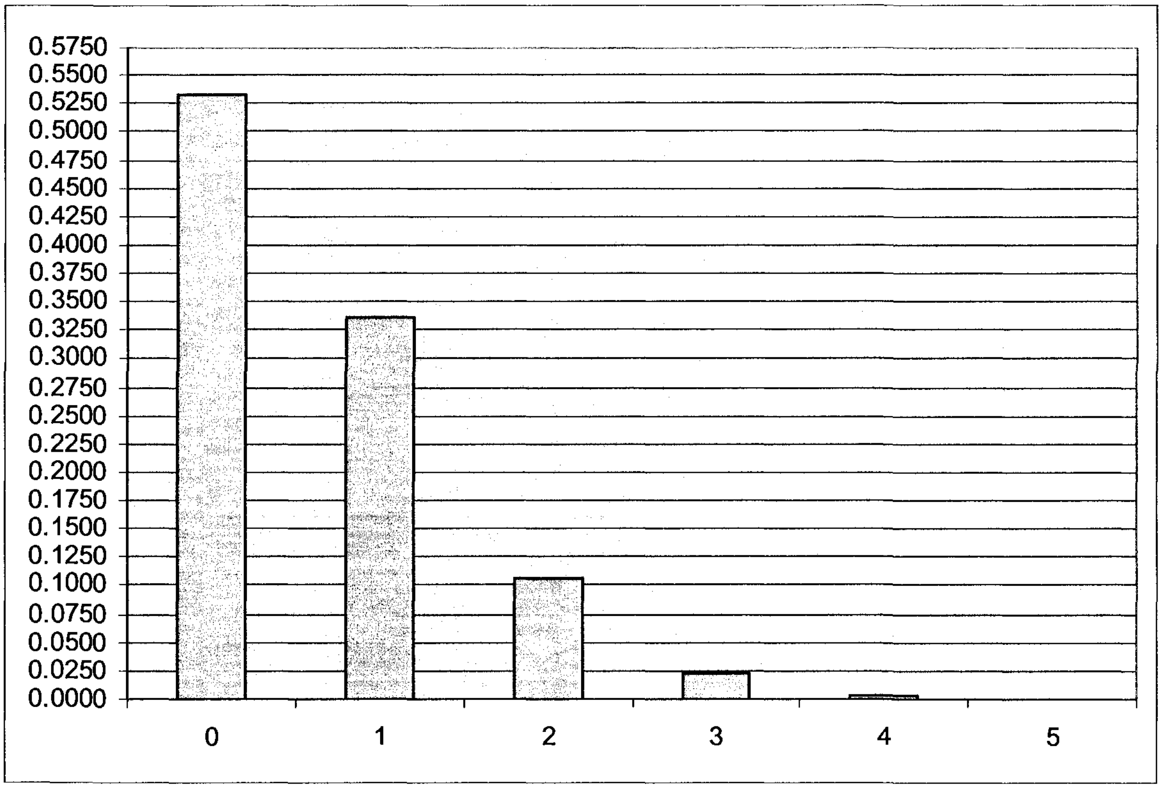


Figure 29 Poisson chart for Main and Charleston

[3] This value does not show on the chart – it is de minimis.

The intersection of Maryland and Tropicana for 2004

Table 14 Total number of accidents by week for 1 year for Maryland & Tropicana

Week #	# of accidents		Week # (con't)	# of accidents (con't)		Week # (con't)	# of accidents (con't)
1	0		19	1		36	2
2	1		20	1		37	1
3	2		21	1		38	1
4	0		22	0		39	3
5	1		23	3		40	1
6	2		24	2		41	1
7	0		25	1		42	0
8	4		26	1		43	1
9	1		27	0		44	4
10	0		28	0		45	0
11	3		29	2		46	0
12	4		30	1		47	3
13	0		31	2		48	0
14	1		32	0		49	1
15	0		33	2		50	1
16	2		34	0		51	2
17	1		35	1		52	1
18	1						

Table 15 There were a total of 15 weeks without an accident for Maryland & Tropicana

Mean	Variance	Lambda (mean+var/2)	Count = 0	Count > 0
1.2115	1.2681	1.2398	15	37

Table 16 The maximum number of accidents reported in one week was 4 for Maryland and Tropicana

Observed	Count	Count / 52	
count = 0	15	0.2885	*The # of weeks with 0 accidents*
count = 1	21	0.4038	*The # of weeks with 1 accident*
count = 2	9	0.1731	*The # of weeks with 2 accidents*
count = 3	4	0.0769	*The # of weeks with 3 accidents*

count = 4	3	0.0577	*The # of weeks with 4 accidents*
Total	*52*	*1.0000*	

Table 17 Poisson distribution for Maryland & Tropicana

Theoretical	Poisson		
X	P(X) /Lambda	P(X) /Mean	P(X) * 52
0	0.2894	0.2977	15.0507
1	0.3588	0.3607	18.6601
2	0.2225	0.2185	11.5676
3	0.0919	0.0882	4.7806
4	0.0285	0.0267	1.4818
5	0.0071	0.0065	0.3674
6	0.0015	0.0013	0.0759
7	0.0003[4]	0.0002	0.0134
			52.00

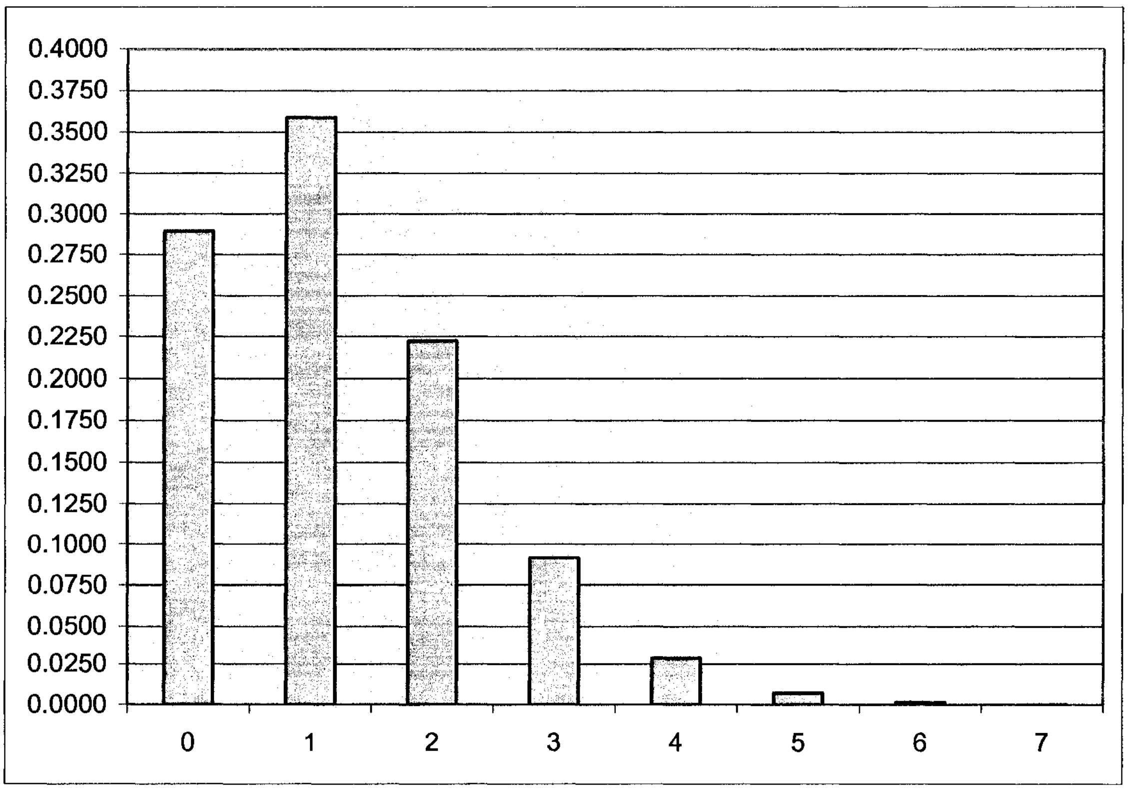


Figure 30 Poisson chart for Maryland and Tropicana

[4] This value does not show on the chart – it is de minimis.

The intersection of Martin Luther King and Cheyenne for 2004

Table 18 Total number of accidents by week for one year for MLK and Cheyenne

Week #	# of accidents		Week # (con't)	# of accidents (con't)		Week # (con't)	# of accidents (con't)
1	1		19	4		36	1
2	0		20	1		37	1
3	2		21	1		38	1
4	0		22	1		39	3
5	1		23	2		40	0
6	1		24	1		41	0
7	2		25	3		42	2
8	1		26	0		43	1
9	3		27	0		44	2
10	0		28	1		45	0
11	0		29	0		46	0
12	2		30	3		47	0
13	1		31	1		48	1
14	3		32	2		49	1
15	0		33	2		50	0
16	0		34	1		51	0
17	0		35	1		52	1
18	1						

Table 19 There were a total of 17 weeks without an accident for MLK and Cheyenne

Mean	Variance	Lambda (mean+var/2)	Count = 0	Count > 0
1.0769	1.0528	1.0649	17	35

Table 20 The maximum number of accidents reported in one week was 4 for MLK and Cheyenne

Observed	Count	Count / 52	
count = 0	17	0.3269	*The # of weeks with 0 accidents*

count = 1	21	0.4038	*The # of weeks with 1 accident*
count = 2	8	0.1538	*The # of weeks with 2 accidents*
count = 3	5	0.0962	*The # of weeks with 3 accidents*
count = 4	1	0.0192	*The # of weeks with 4 accidents*
Total	*52*	*1.0000*	

Table 21 Poisson distribution for MLK and Cheyenne

Theoretical	Poisson		
X	P(X) /Lambda	P(X) /Mean	P(X) * 52
0	0.3448	0.3406	17.9284
1	0.3671	0.3668	19.0912
2	0.1955	0.1975	10.1647
3	0.0694	0.0709	3.6080
4	0.0185	0.0191	0.9605
5	0.0039	0.0041	0.2046
6	0.0007	0.0007	0.0363
7	0.0001[5]	0.0001	0.0055
			52.00

[5] This value does not show on the chart -- it is de minimis.

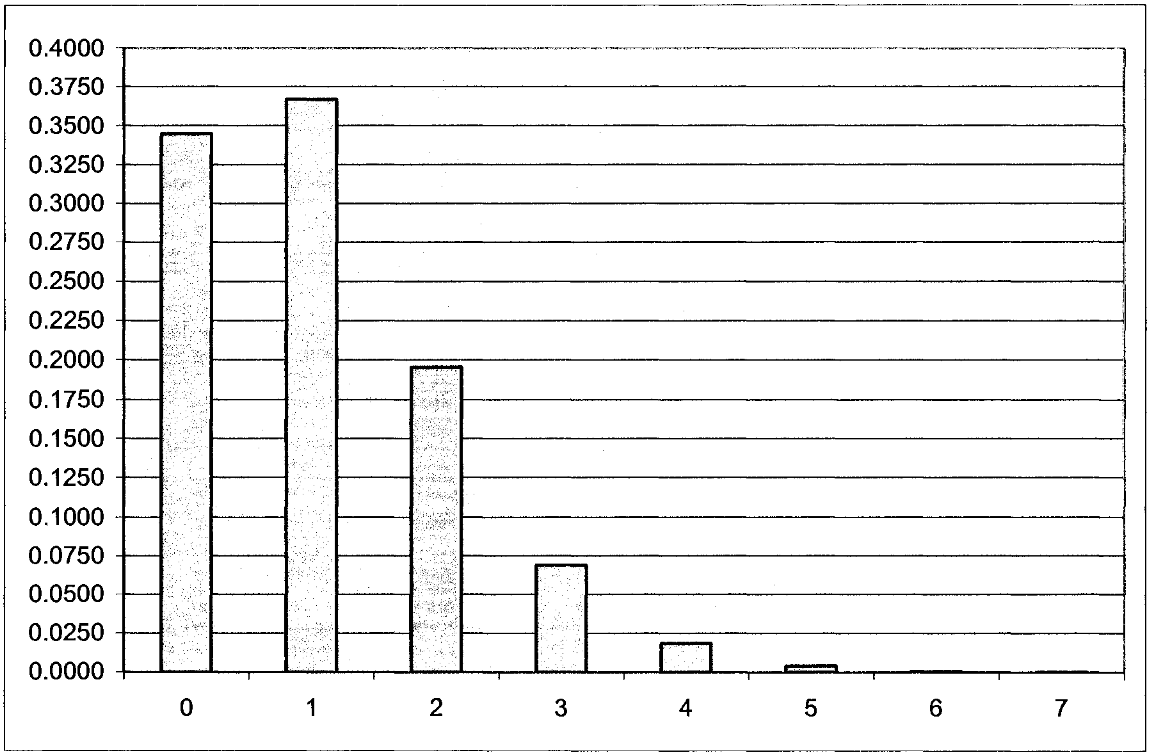


Figure 31 Poisson chart for Martin Luther King and Cheyenne

The intersection of Martin Luther King and Craig for 2002

Table 22 Total number of accidents by week for one year for MLK and Craig

Week #	# of accidents		Week # (con't)	# of accidents (con't)		Week # (con't)	# of accidents (con't)
1	3		19	1		36	3
2	2		20	1		37	1
3	1		21	1		38	0
4	1		22	4		39	0
5	2		23	1		40	1
6	1		24	1		41	1
7	2		25	0		42	0
8	0		26	2		43	1
9	0		27	1		44	0
10	1		28	5		45	0
11	1		29	1		46	1
12	3		30	0		47	1
13	1		31	0		48	1
14	0		32	1		49	0
15	1		33	2		50	0
16	2		34	1		51	1
17	0		35	2		52	1
18	1						

Table 23 There were a total of 14 weeks without an accident for MLK and Craig

Mean	Variance	Lambda (mean+var/2)	Count = 0	Count > 0
1.1154	1.1237	1.1195	14	38

Table 24 The maximum number of accidents reported in one week was 5 for MLK and Craig

Observed	Count	Count / 52	
count = 0	14	0.2692	*The # of weeks with 0 accidents*

count = 1	26	0.5000	*The # of weeks with 1 accident*
count = 2	7	0.1346	*The # of weeks with 2 accidents*
count = 3	3	0.0577	*The # of weeks with 3 accidents*
count = 4	1	0.0192	*The # of weeks with 4 accidents*
count = 5	1	0.0192	*The # of weeks with 5 accidents*
Total	*52*	*1.0000*	

Table 25 Poisson distribution for Martin Luther King and Craig

Theoretical	Poisson		
X	P(X) /Lambda	P(X) /Mean	P(X) * 52
0	0.3264	0.3278	16.9745
1	0.3655	0.3656	19.0035
2	0.2046	0.2039	10.6375
3	0.0763	0.0758	3.9697
4	0.0214	0.0211	1.1110
5	0.0048	0.0047	0.2488
6	0.0009	0.0009	0.0464
7	0.0001[6]	0.0001	0.0074
			52.00

[6] This value does not show on the chart – it is de minimis.

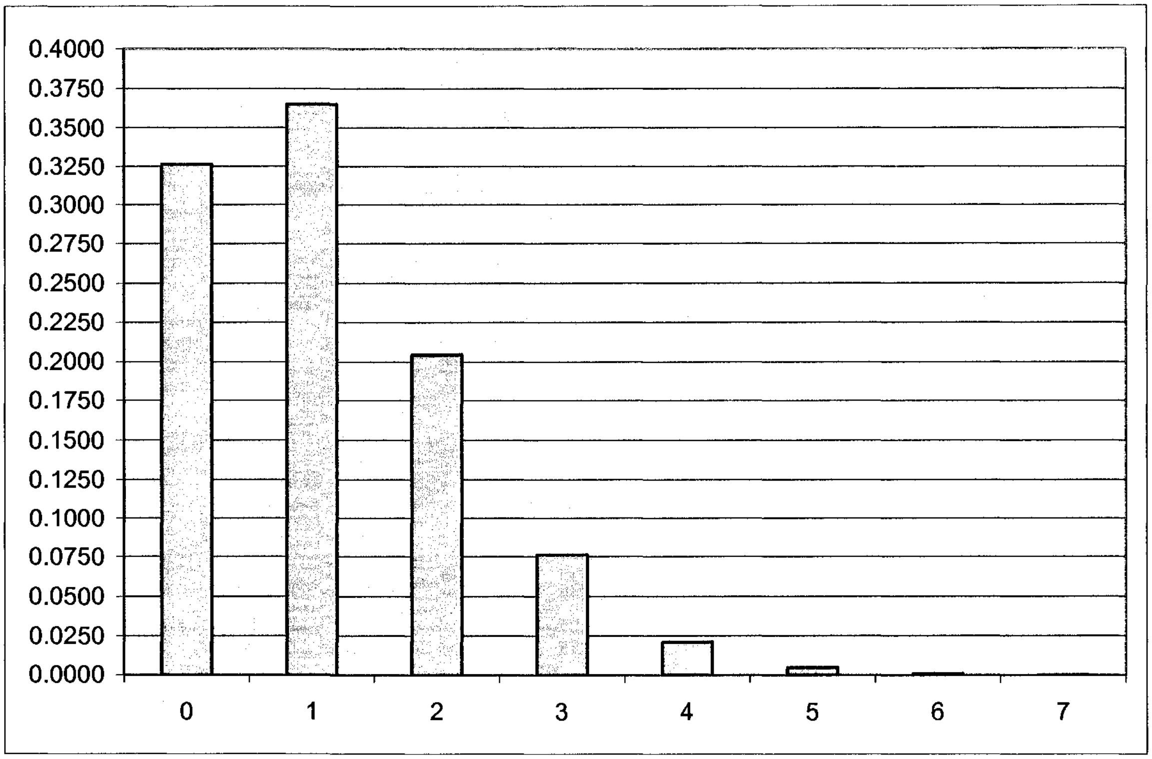


Figure 32 Poisson chart for Martin Luther King and Craig

The intersection of Sahara and Durango for 2006

Table 26 Total number of accidents by week for one year for Sahara and Durango

Week #	# of accidents		Week # (con't)	# of accidents (con't)		Week # (con't)	# of accidents (con't)
1	1		19	1		36	3
2	0		20	0		37	2
3	2		21	1		38	2
4	4		22	2		39	2
5	4		23	1		40	3
6	1		24	1		41	4
7	0		25	2		42	2
8	2		26	2		43	2
9	0		27	1		44	2
10	0		28	1		45	1
11	1		29	2		46	3
12	0		30	2		47	0
13	2		31	2		48	0
14	3		32	0		49	0
15	1		33	1		50	0
16	4		34	2		51	0
17	0		35	2		52	0
18	0						

Table 27 There were a total of 15 weeks without an accident for Sahara & Durango

Mean	Variance	Lambda (mean+var/2)	Count = 0	Count > 0
1.4231	1.4646	1.4438	15	37

Table 28 The maximum number of accidents reported in one week was 4 for Sahara and Durango

Observed	Count	Count / 52	
count = 0	15	0.2885	*The # of weeks with 0 accidents*

count = 1	12	0.2308	*The # of weeks with 1 accident*
count = 2	17	0.3269	*The # of weeks with 2 accidents*
count = 3	4	0.0769	*The # of weeks with 3 accidents*
count = 4	4	0.0769	*The # of weeks with 4 accidents*
Total	*52*	*1.0000*	

Table 29 Poisson distribution for Sahara and Durango

Theoretical	Poisson		
X	P(X) /Lambda	P(X) /Mean	P(X) * 52
0	0.2360	0.2410	12.2733
1	0.3408	0.3429	17.7204
2	0.2460	0.2440	12.7925
3	0.1184	0.1157	6.1567
4	0.0427	0.0412	2.2223
5	0.0123	0.0117	0.6417
6	0.0030	0.0028	0.1544
7	0.0006	0.0006	0.0319
8	0.0001[7]	0.0001	0.0057
			52.00

[7] This value does not show on the chart – it is de minimis.

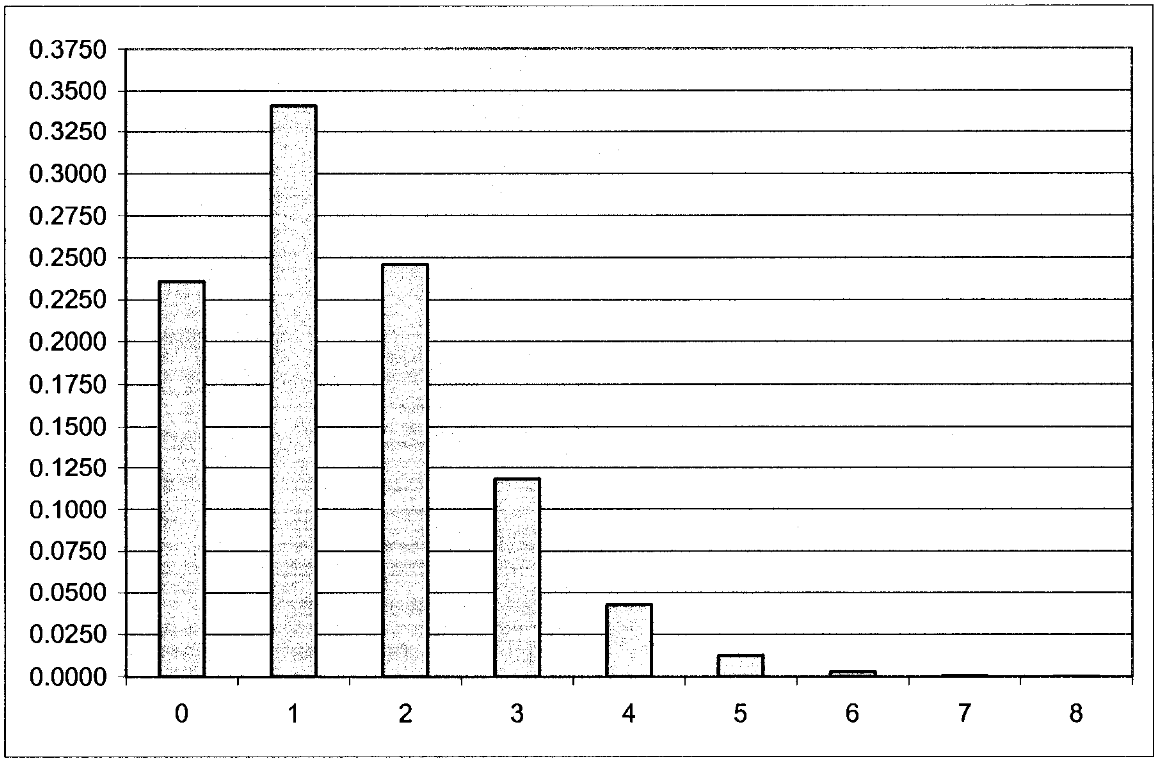


Figure 33 Poisson chart for Sahara and Durango

The intersection of Warm Springs and Rainbow for 2006

Table 30 Total number of accidents by week for one year for Warm Springs and Rainbow

Week #	# of accidents		Week # (con't)	# of accidents (con't)		Week # (con't)	# of accidents (con't)
1	0		19	1		36	1
2	2		20	0		37	2
3	0		21	1		38	0
4	1		22	2		39	3
5	2		23	1		40	0
6	3		24	1		41	0
7	1		25	0		42	1
8	0		26	0		43	1
9	0		27	0		44	0
10	0		28	0		45	0
11	1		29	0		46	0
12	0		30	0		47	1
13	0		31	0		48	1
14	1		32	1		49	0
15	0		33	0		50	0
16	1		34	1		51	0
17	0		35	1		52	0
18	1						

Table 31 There were a total of 28 weeks without an accident for Warm Springs and Rainbow

Mean	Variance	Lambda (mean+var/2)	Count = 0	Count > 0
0.6154	0.6335	0.6244	28	24

Table 32 The maximum number of accidents reported in one week was 3 for Warm Springs and Rainbow

Observed	Count	Count / 52	
count = 0	28	0.5385	*The # of weeks with 0 accidents*
count = 1	18	0.3462	*The # of weeks with 1 accident*
count = 2	4	0.0769	*The # of weeks with 2 accidents*
count = 3	2	0.0385	*The # of weeks with 3 accidents*
Total	*52*	*1.0000*	

Table 33 Poisson distribution for Warm Springs and Rainbow

Theoretical	Poisson		
X	P(X) /Lambda	P(X) /Mean	P(X) * 52
0	0.5356	0.5404	27.8493
1	0.3344	0.3326	17.3901
2	0.1044	0.1023	5.4295
3	0.0217	0.0210	1.1301
4	0.0034	0.0032	0.1764
5	0.0004[8]	0.0004	0.0220
			52.00

[8] This value does not show on the chart – it is de minimis.

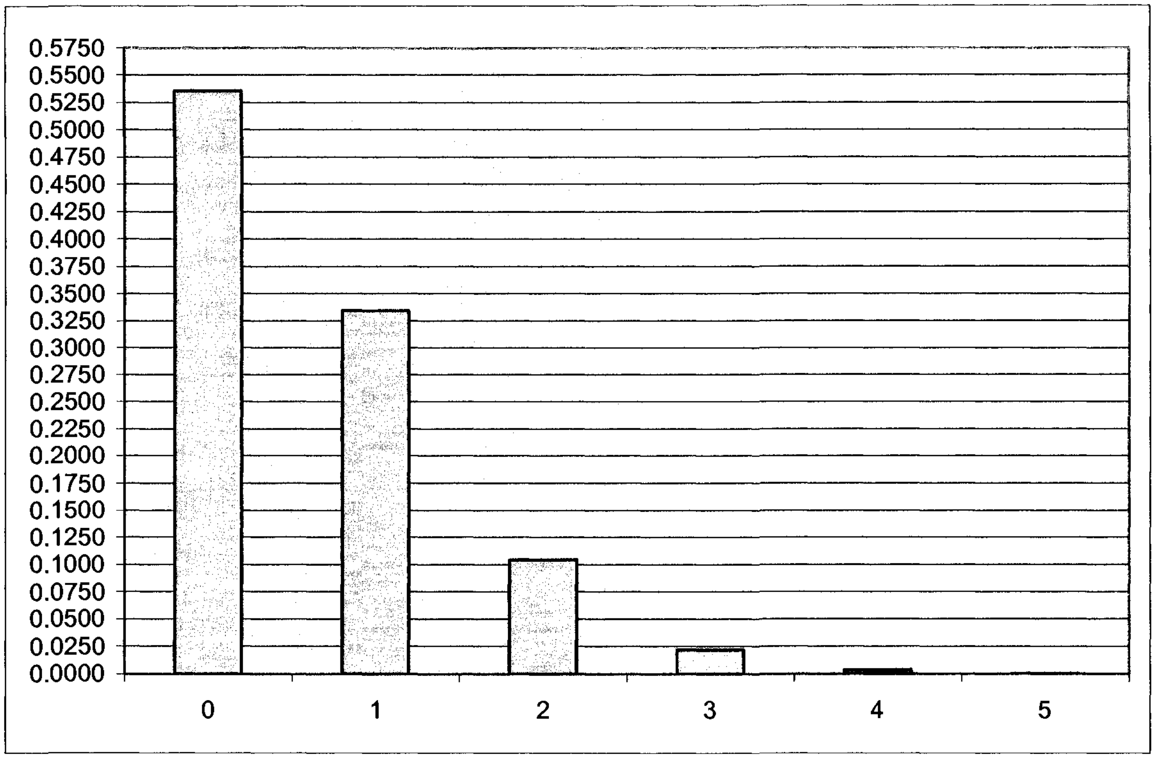


Figure 34 Poisson chart for Warm Springs and Rainbow

The intersection of Jones and Sahara for 2002

Table 34 Total number of accidents by week for one year for Jones and Sahara

Week #	# of accidents		Week # (con't)	# of accidents (con't)		Week # (con't)	# of accidents (con't)
1	0		19	1		36	0
2	1		20	0		37	2
3	0		21	1		38	0
4	0		22	0		39	0
5	3		23	1		40	1
6	1		24	0		41	0
7	3		25	2		42	1
8	3		26	0		43	2
9	2		27	1		44	0
10	2		28	1		45	2
11	1		29	1		46	4
12	2		30	0		47	0
13	0		31	2		48	3
14	1		32	1		49	1
15	3		33	2		50	2
16	3		34	3		51	2
17	1		35	0		52	4
18	2						

Table 35 There were a total of 16 weeks without an accident for Jones and Sahara

Mean	Variance	Lambda (mean+var/2)	Count = 0	Count > 0
1.3077	1.3544	1.3311	16	36

Table 36 The maximum number of accidents reported in one week was four for Jones and Sahara

Observed	Count	Count / 52	
count = 0	16	0.3077	*The # of weeks with 0 accidents*

count = 1	15	0.2885	*The # of weeks with 1 accident*
count = 2	12	0.2308	*The # of weeks with 2 accidents*
count = 3	7	0.1346	*The # of weeks with 3 accidents*
count = 4	2	0.0385	*The # of weeks with 4 accidents*
Total	*52*	*1.0000*	

Table 37 Poisson distribution for Jones and Sahara

Theoretical	Poisson		
X	P(X) /Lambda	P(X) /Mean	P(X) * 52
0	0.2642	0.2704	13.7381
1	0.3517	0.3537	18.2864
2	0.2340	0.2312	12.1702
3	0.1038	0.1008	5.3998
4	0.0346	0.0330	1.7969
5	0.0092	0.0086	0.4784
6	0.0020	0.0019	0.1061
7	0.0004[9]	0.0004	0.0202
			52.00

[9] This value does not show on the chart – it is de minimis.

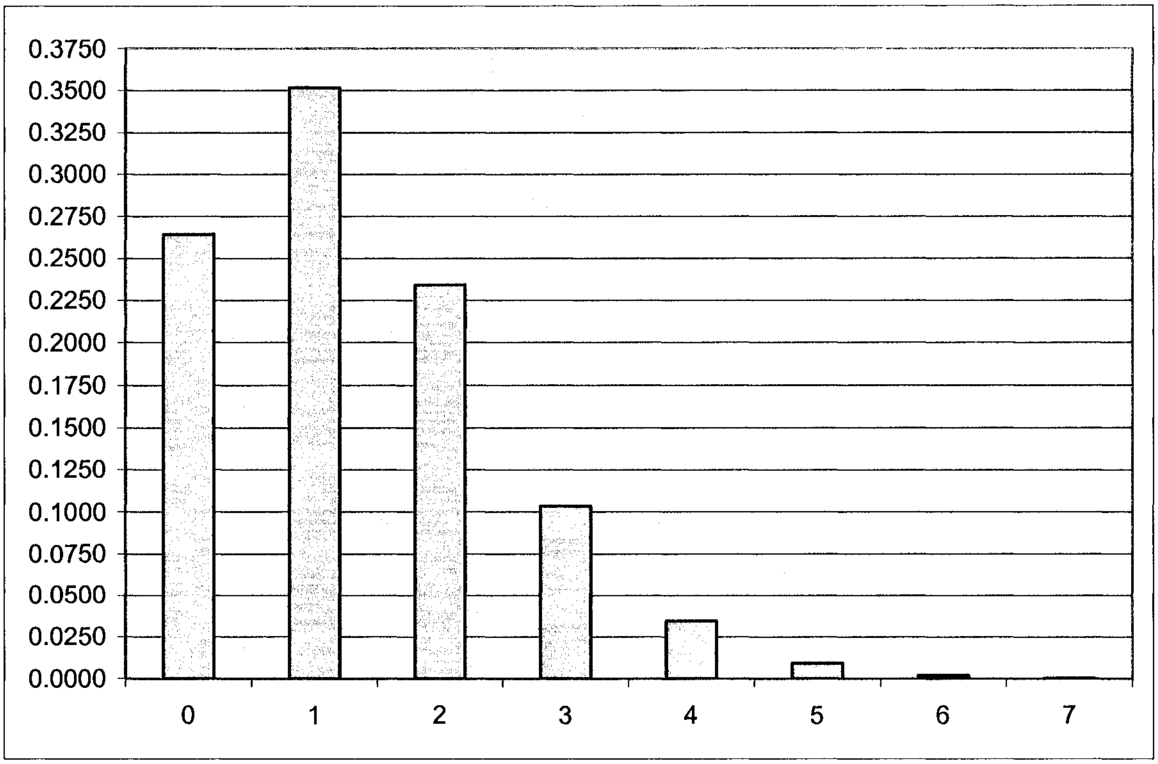


Figure 35 Poisson chart for Jones and Sahara

The intersection of Flamingo and Maryland for 2002

Table 38 Total number of accidents by week for one year for Flamingo & Maryland

Week #	# of accidents		Week # (con't)	# of accidents (con't)		Week # (con't)	# of accidents (con't)
1	1		19	1		36	5
2	3		20	3		37	3
3	2		21	3		38	6
4	3		22	4		39	2
5	5		23	1		40	2
6	3		24	5		41	2
7	0		25	3		42	4
8	3		26	2		43	2
9	4		27	0		44	6
10	1		28	1		45	0
11	7		29	3		46	2
12	2		30	3		47	4
13	2		31	4		48	2
14	4		32	7		49	1
15	7		33	0		50	1
16	3		34	0		51	4
17	5		35	6		52	7
18	4						

Table 39 There were a total of five weeks without an accident for Flamingo and Maryland

Mean	Variance	Lambda (mean+var/2)	Count = 0	Count > 0
3.0385	3.8808	3.4597	5	47

Table 40 The maximum number of accidents reported in one week was 7 for Flamingo and Maryland

Observed	Count	Count / 52	

count = 0	5	0.0962	*The # of weeks with 0 accidents*
count = 1	7	0.1346	*The # of weeks with 1 accident*
count = 2	10	0.1923	*The # of weeks with 2 accidents*
count = 3	11	0.2115	*The # of weeks with 3 accidents*
count = 4	8	0.1538	*The # of weeks with 4 accidents*
count = 5	4	0.0769	*The # of weeks with 5 accidents*
count = 6	3	0.0577	*The # of weeks with 6 accidents*
count = 7	4	0.0769	*The # of weeks with 7 accidents*
Total	*52*	*1.0000*	

Table 41 Poisson distribution for Flamingo and Maryland

Theoretical	Poisson		
X	P(X) /Lambda	P(X) /Mean	P(X) * 52
0	0.0314	0.0479	1.6349
1	0.1088	0.1456	5.6562
2	0.1882	0.2212	9.7843
3	0.2170	0.2240	11.2834
4	0.1877	0.1701	9.7592
5	0.1299	0.1034	6.7527
6	0.0749	0.0524	3.8937
7	0.0370	0.0227	1.9244
8	0.0160	0.0086	0.8322
9	0.0062	0.0029	0.3199
10	0.0021	0.0009	0.1107
11	0.0007	0.0002	0.0348
12	0.0002[10]	0.0001	0.0100
			52.00

[10] This value does not show on the chart – it is de minimis.

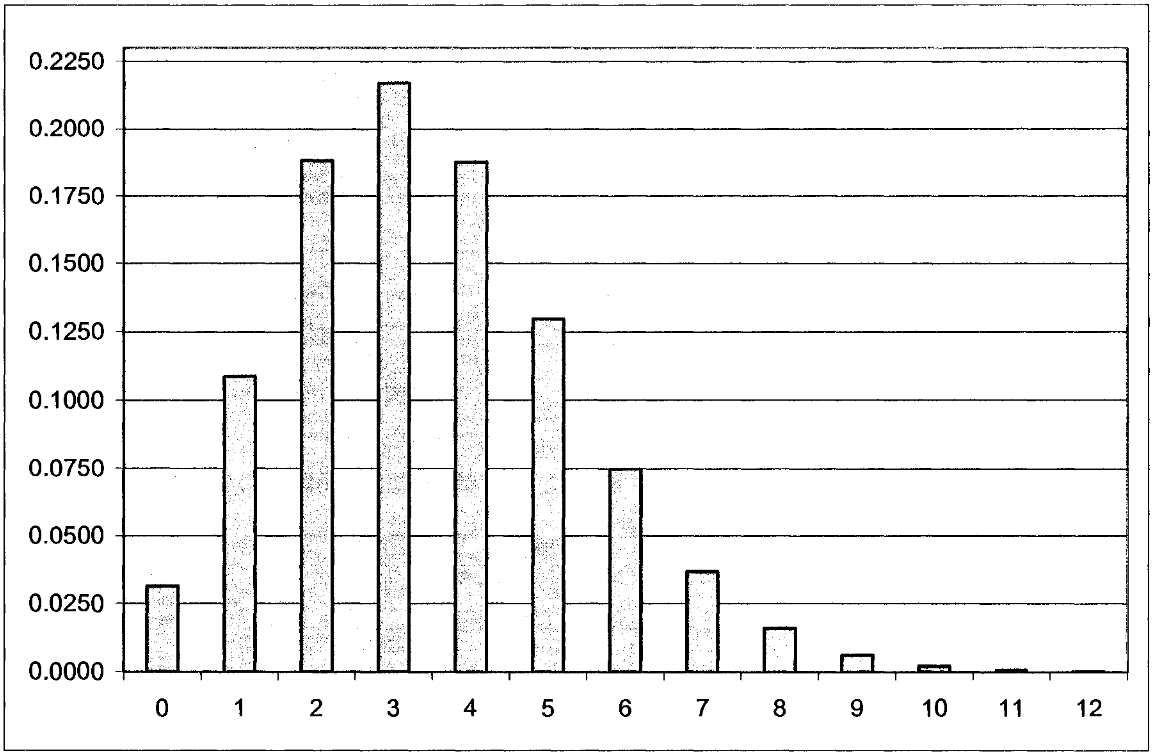


Figure 36 Poisson chart for Flamingo and Maryland

The intersection of Cheyenne and Rancho for 2004

Table 42 Total number of accidents by week for one year for Cheyenne and Rancho

Week #	# of accidents		Week # (con't)	# of accidents (con't)		Week # (con't)	# of accidents (con't)
1	0		19	1		36	0
2	1		20	0		37	0
3	1		21	0		38	1
4	0		22	0		39	0
5	0		23	0		40	2
6	0		24	0		41	2
7	1		25	0		42	1
8	2		26	0		43	0
9	0		27	3		44	0
10	1		28	2		45	1
11	0		29	0		46	0
12	2		30	1		47	1
13	1		31	1		48	0
14	2		32	0		49	0
15	2		33	0		50	0
16	1		34	0		51	0
17	1		35	2		52	0
18	1						

Table 43 There were a total of 28 weeks without an accident for Cheyenne and Rancho

Mean	Variance	Lambda (mean+var/2)	Count = 0	Count > 0
0.6539	0.6621	0.6580	28	24

Table 44 The maximum number of accidents reported in 1 week was three for Cheyenne and Rancho

Observed	Count	Count / 52	
count = 0	28	0.5385	*The # of weeks with 0 accidents*

count = 1	15	0.2885	*The # of weeks with 1 accident*
count = 2	8	0.1538	*The # of weeks with 2 accidents*
count = 3	1	0.0192	*The # of weeks with 3 accidents*
Total	*52*	*1.0000*	

Table 45 Poisson distribution for Cheyenne and Rancho

Theoretical	Poisson		
X	P(X) /Lambda	P(X) /Mean	P(X) * 52
0	0.5179	0.5200	26.9302
1	0.3408	0.3400	17.7199
2	0.1121	0.1112	5.8298
3	0.0246	0.0242	1.2787
4	0.0040	0.0040	0.2103
5	0.0005[11]	0.0005	0.0277
			52.00

[11] This value does not show on the chart – it is de minimis.

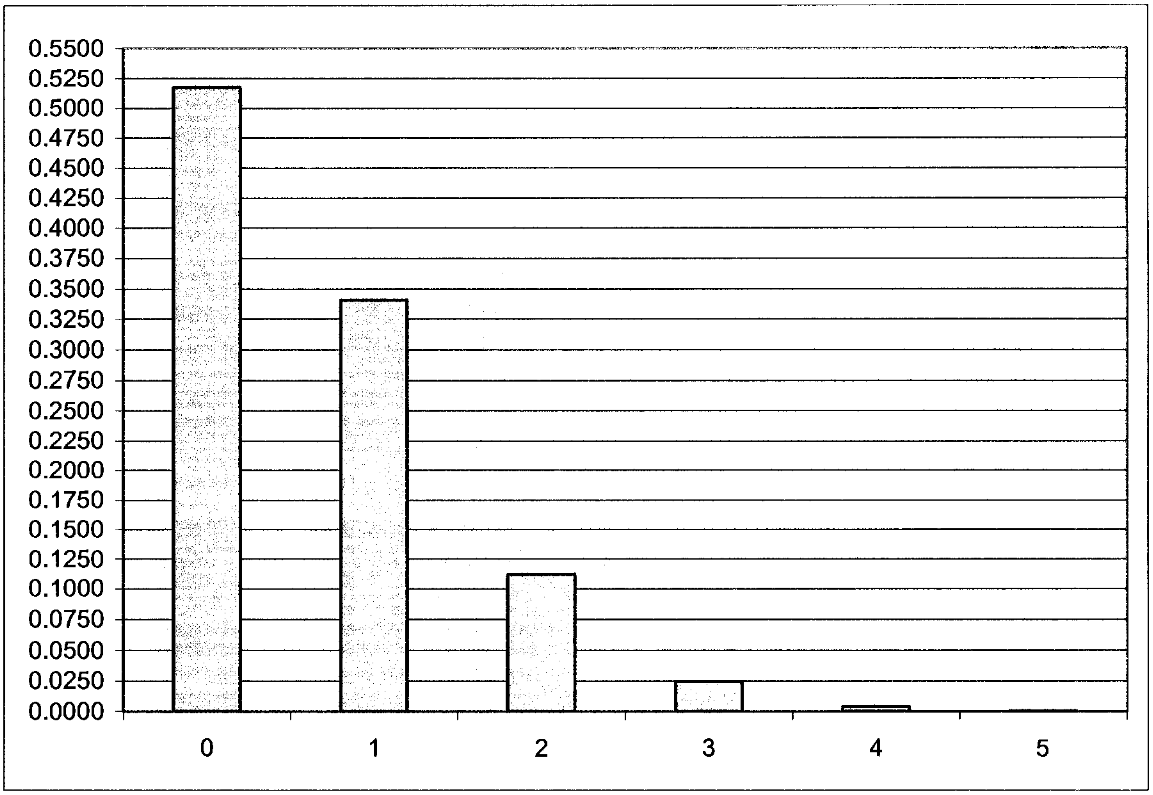


Figure 37 Poisson chart for Cheyenne and Rancho

The intersection of Buffalo and Lone Mountain for 2006

Table 46 Total number of accidents by week for one year for Buffalo and Lone Mountain

Week #	# of accidents		Week # (con't)	# of accidents (con't)		Week # (con't)	# of accidents (con't)
1	0		19	1		36	1
2	0		20	0		37	0
3	0		21	0		38	0
4	0		22	0		39	0
5	1		23	0		40	0
6	1		24	0		41	1
7	0		25	0		42	0
8	0		26	0		43	0
9	0		27	0		44	0
10	0		28	0		45	0
11	0		29	0		46	0
12	0		30	0		47	2
13	0		31	1		48	0
14	0		32	0		49	0
15	0		33	0		50	0
16	1		34	0		51	0
17	0		35	0		52	0
18	0						

Table 47 There were a total of 44 weeks without an accident for Buffalo and Lone Mountain

Mean	Variance	Lambda (mean+var/2)	Count = 0	Count > 0
0.1731	0.1851	0.1791	44	8

Table 48 The maximum number of accidents reported in one week was 2 for Buffalo and Lone Mountain

Observed	Count	Count / 52	
count = 0	44	0.8462	*The # of weeks with 0 accidents*
count = 1	7	0.1346	*The # of weeks with 1 accident*
count = 2	1	0.0192	*The # of weeks with 2 accidents*
Total	*52*	*1.0000*	

Table 49 Poisson distribution for Buffalo and Lone Mountain

Theoretical	Poisson		
X	P(X) /Lambda	P(X) /Mean	P(X) * 52
0	0.8360	0.8411	43.4727
1	0.1497	0.1456	7.7864
2	0.0134	0.0126	0.6973
3	0.0008[12]	0.0007	0.0416
			52.00

[12] This value does not show on the chart – it is de minimis.

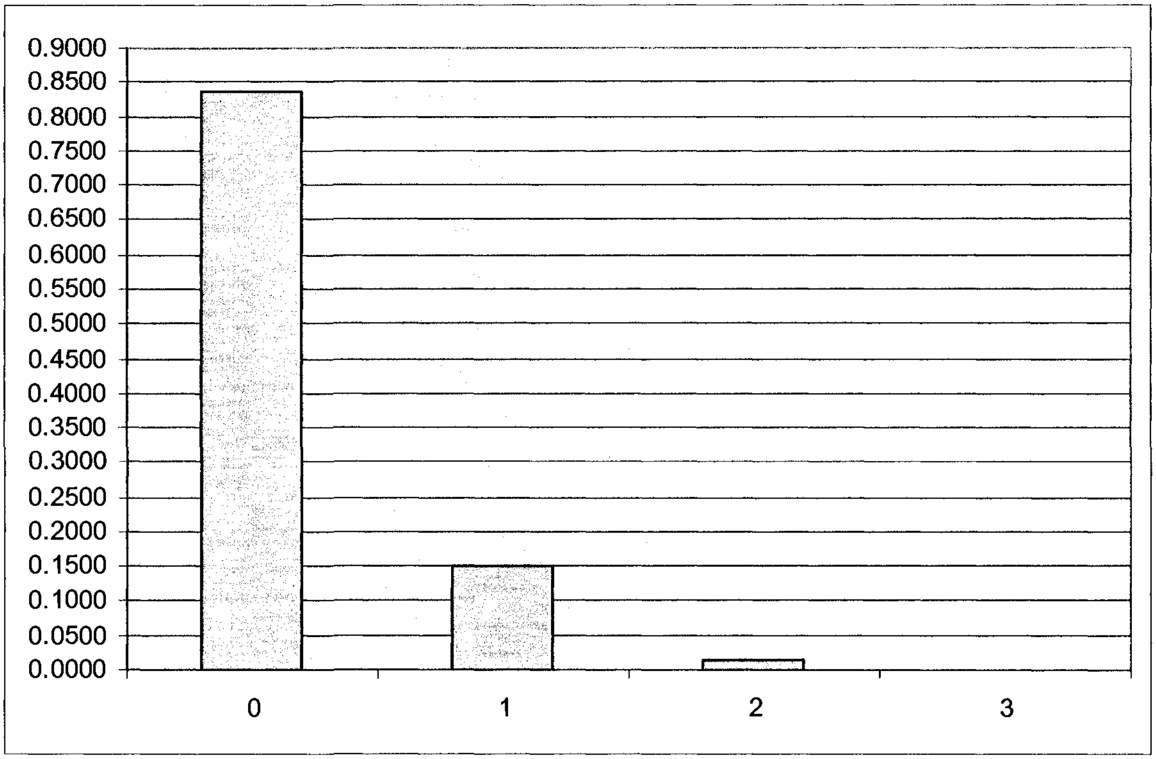


Figure 38 Poisson chart for Buffalo and Lone Mountain

The intersection of Lamb and Craig for 2005

Table 50 Total number of accidents by week for one year for Lamb and Craig

Week #	# of accidents		Week # (con't)	# of accidents (con't)		Week # (con't)	# of accidents (con't)
1	0		19	3		36	0
2	0		20	0		37	1
3	1		21	0		38	0
4	0		22	0		39	0
5	1		23	0		40	0
6	0		24	0		41	0
7	0		25	0		42	0
8	0		26	1		43	2
9	0		27	2		44	0
10	2		28	0		45	1
11	2		29	1		46	0
12	0		30	1		47	0
13	1		31	2		48	0
14	2		32	0		49	0
15	1		33	0		50	1
16	1		34	0		51	0
17	0		35	0		52	0
18	0						

Table 51 There were a total of 34 weeks without an accident for Lamb and Craig

Mean	Variance	Lambda (mean+var/2)	Count = 0	Count > 0
0.5000	0.6078	0.5539	34	18

Table 52 The maximum number of accidents reported in one week was three for Lamb and Craig

Observed	Count	Count / 52	
count = 0	34	0.6538	*The # of weeks with 0 accidents*

count = 1	11	0.2115	*The # of weeks with 1 accident*
count = 2	6	0.1154	*The # of weeks with 2 accidents*
count = 3	1	0.0192	*The # of weeks with 3 accidents*
Total	*52*	*1.0000*	

Table 53 Poisson distribution for Lamb and Craig

Theoretical	Poisson		
X	P(X) /Lambda	P(X) /Mean	P(X) * 52
0	0.5747	0.6065	29.8840
1	0.3183	0.3033	16.5534
2	0.0882	0.0758	4.5846
3	0.0163	0.0126	0.8465
4	0.0023	0.0016	0.1172
5	0.0002[13]	0.0002	0.0130
			52.00

[13] This value does not show on the chart – it is de minimis.

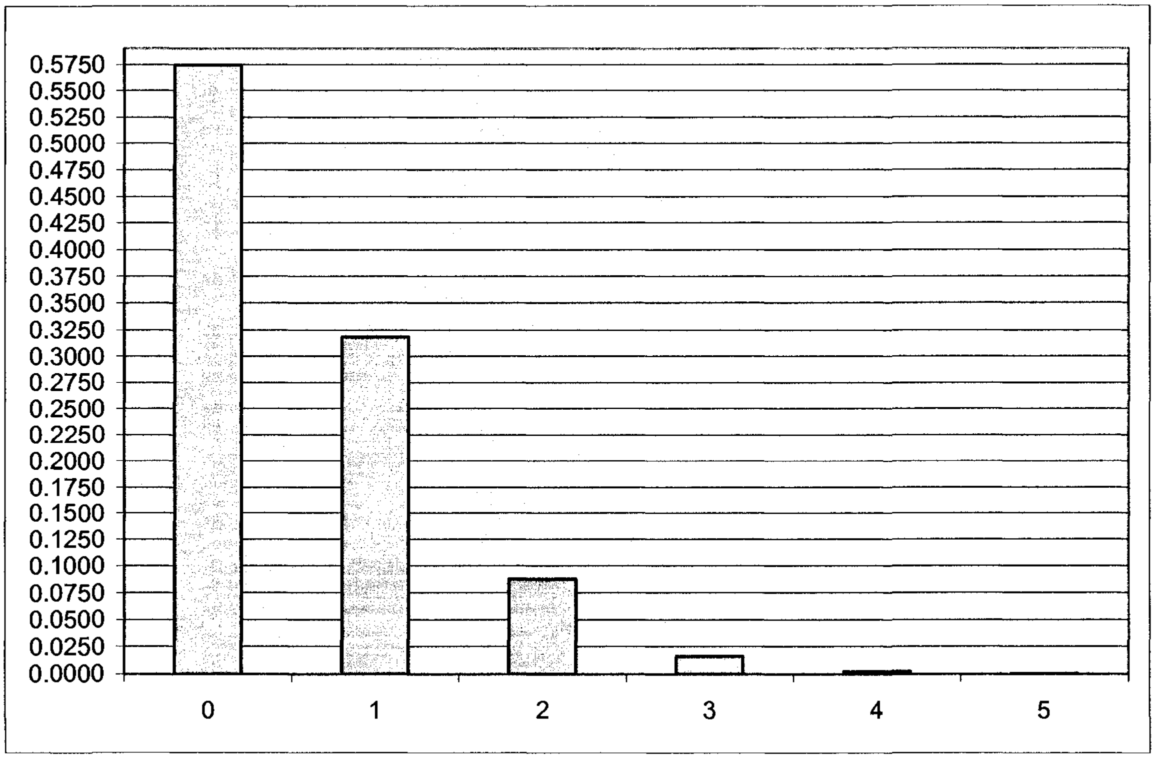


Figure 39 Poisson chart for Lamb and Craig

APPENDIX B

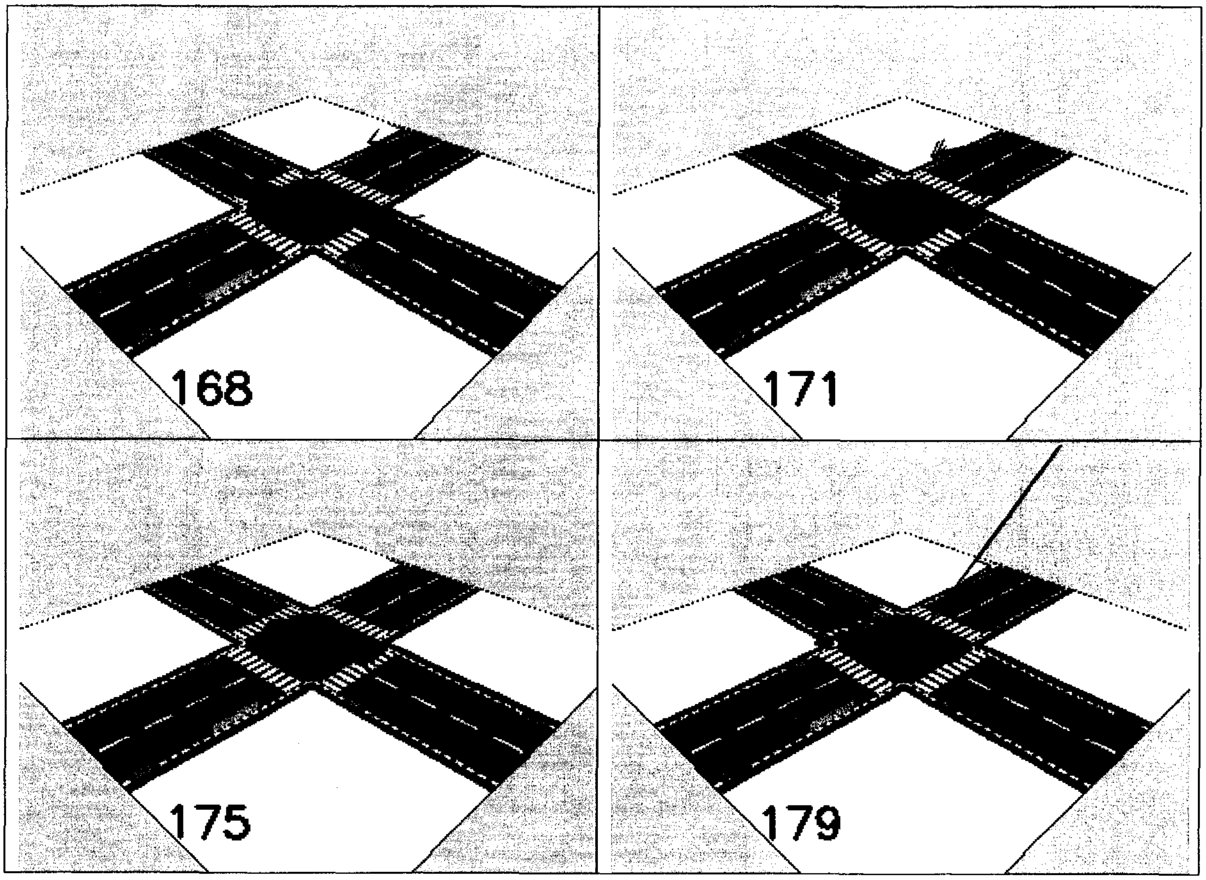
SUMMARY OF EDGE DETECTORS

Table 54 Summary of Edge Detectors

Edge Detector	Short Description
Canny	uses smoothing before edge detection and thresholding
Frei	calculates the Frei-Chen edge operator using only the row and column filters
Krish[52]	performs convolution with 8 masks calculating gradients
Laplace	finds zero crossings of the second derivative of the image intensity
Marr-Hildreth[52]	performs two convolutions with Laplacian of Gaussian and then detects the zero crossings
Prewitt	calculates the Prewitt compass gradient filters and returns the largest filter response
Roberts	calculates the square root of the magnitude squared of the convolution with the Robert's row and column edge detectors
Sobel	uses convolutions with row and column edge gradient masks

APPENDIX C

EXPERIMENTAL PROGRAM RESULTS



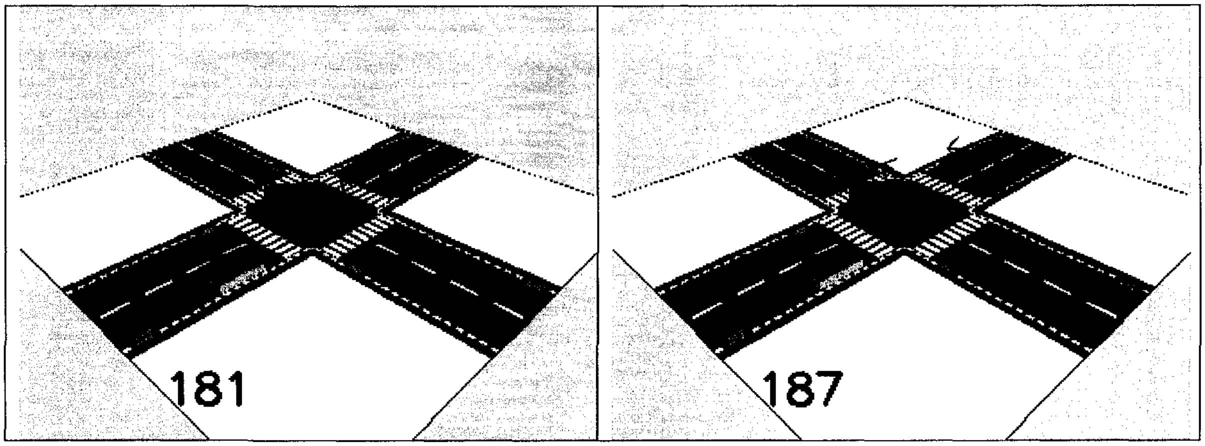


Figure 40 Example of using Optical Flow to detect 'backward' movement in the intersection

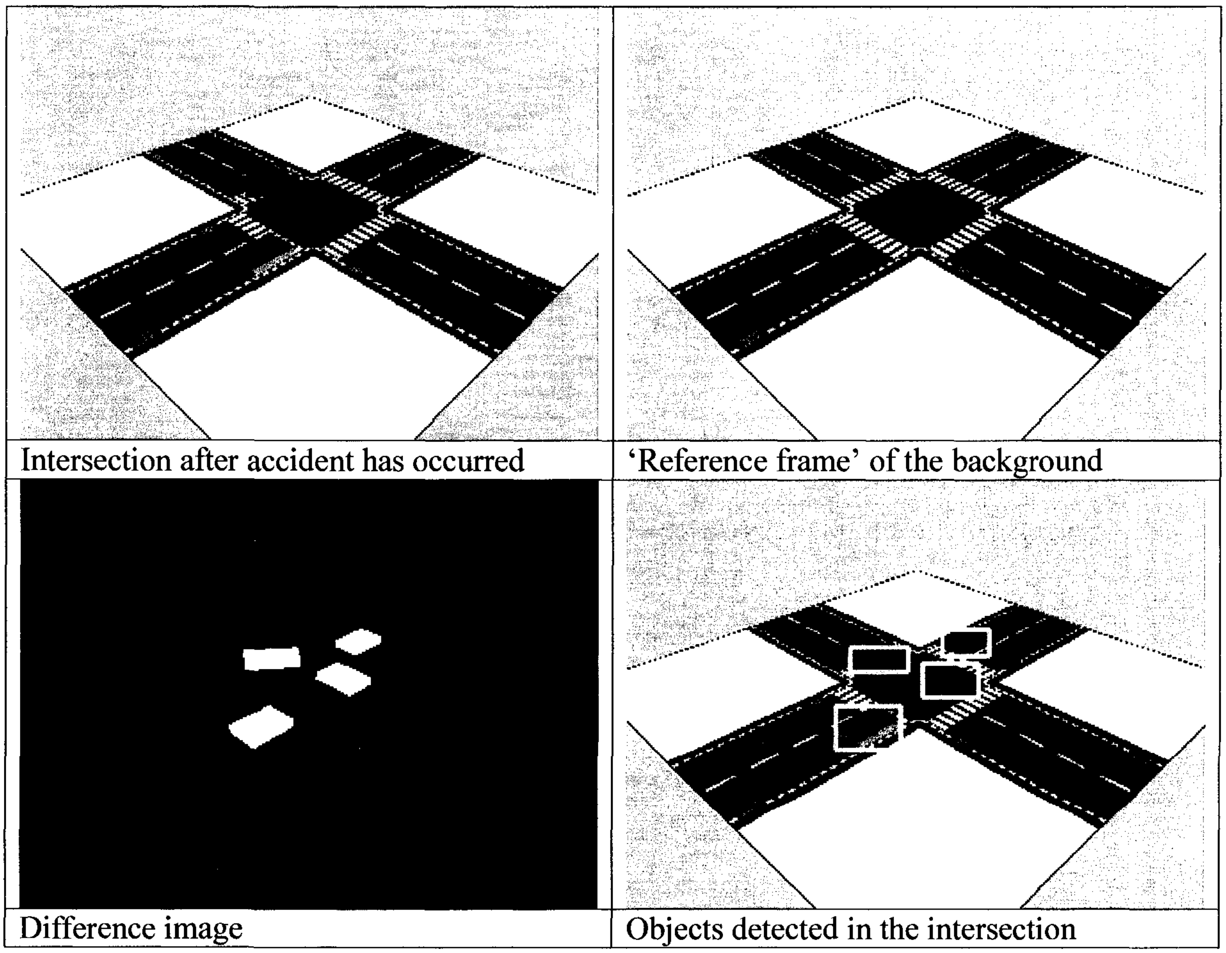


Figure 41 Example of using 'reference frame' to detect objects that have not left the intersection and have stopped moving

REFERENCES

[1] Collins, R.T., Lipton, A.J., and Kanade, T., "Introduction to the Special Section on Video Surveillance", IEEE Transactions on PAMI, Volume 22(8), August 2000.

[2] Chen, T.P., Haussecker, H., Bovyrin, A., Belenov, R., Rodyushkin, K., Kuranov, A., Eruhimov, V., "Computer Vision Workload analysis: Case Study of Video Surveillance Systems", Intel Technology Journal, Volume 9, Issue 2, 2005.

[3] Aubert, D., Bouzar, S., Lenoir, F., Blosseville, J.M."Automatic Vehicle Queue Measurement at Intersection Using Image Processing" IEE Road Traffic Monitoring and Control, Conference Publication no. 422, pp. 100-104 (1996).

[4] Fathy, M., Siyal, M.Y. . "Real-time image processing approach to measure traffic queue parameters" Vision, Image and Signal Processing, IEE Proceedings, Volume 142, no. 5, pp. 297-303 (1995).

[5] Michalopoulos, P "Vehicle Detection video through image processing: the Autoscope system" IEEE Transactions on Vehicular Technology 40-1 21-29, 1991.

[6] Valera, M.; Velastin, S.A., "Intelligent distributed surveillance systems: a review", Vision, Image and Signal Processing, IEE Proceedings Volume 152, Issue 2, 8 April 2005 Page(s): 192 – 204.

[7] Faugeras, O. "Three dimensional computer vision – a geometric viewpoint", Artificial Intelligence, M.I.T. Press, Cambridge, MA. 1993.

[8] Xu, L., Landabaso, J., Lei B., "Segmentation and tracking of multiple moving objects for intelligent video analysis", BT Technology Journal, Volume 22(3), July 2004.

[9] Cutler, R., Davis, L.S., "Robust real-time periodic motion detection, analysis, and applications", IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 22(8), pp 781-796, 2000.

[10] Ralston, A., Reilly, E.D., Hemmendinger, D., Encyclopedia of Computer Science, 4th edition, Nature Publishing Group, 2000.

[11] Encyclopedia of Science and Technology, Volume 4, 10th edition, McGraw-Hill, 2007.

[12] Hu, W., Tan. T., Wang, L., Maybank, S., "A survey on visual surveillance of object motion and behaviors", IEEE Transactions on Systems, Man and Cybernetics, Part C, Volume 34(3), August 2004.

[13] Intel OpenCV (Open Source Computer Vision Library), http://www.intel.com/technology/computing/opencv/index.htm .

[14] The New Encyclopedia Britannica, Macropedia, Volume 16, 15th edition, 2005.

[15] Stevenson, B., "The Home Book of Proverbs, Maxims, and Familiar Phrases", New York: Macmillan. pg. 2611, 1948.

[16] Picardi, M., Jan, T., "The Industrial Physicist", Vol. 9, No. 1, February/March 2003.

[17] Gupte, S., Masoud, O., Martin, R., Papanikolopoulos, N., "Detection and Classification of Vehicles", IEEE Transactions On Intelligent Transportation Systems, Vol. 3, No.1, pg. 37 – 47, March 2002.

[18] Collins, R.T. et al., "A system for video surveillance and monitoring: VSAM final report", CMU-RI-TR-00-12, Technical Report, Carnegie Mellon University, May 2000.

[19] Mohottala, S., Kagesawa, M., Ikeuchi, K., "Vehicle Class Recognition Using 3D CG Models", 10th World Congress on Intelligent Transport Systems, November 2003.

[20] Yoshida, T., Kagesawa, M., Tomonaka, T., Ikeuchi, K., "Vehicle Recognition with Local-Feature Based Algorithm Using Parallel Vision Board", 8th World Congress on Intelligent Transport Systems, Sydney, 2001.

[21] Matsushita, Y., Murao, M., Kamijo, S., Ikeuchi, K., Sakauchi, M., "Visualization of Traffic Activities at Intersections," ITS, pp. 100-108, September 2001.

[22] Coifman, B., Beymer, D., McLauchlan, P., Malik, J., "A Real-Time Computer Vision System for Vehicle Tracking and Traffic Surveillance", Transportation Research: Part C, Vol. 6, No. 4, pg. 271-288 1998.

[23] Ran, B., Liu, H. X., "Development of Vision-Based Vehicle Detection and Recognition System for Intelligent Vehicles", Transportation Research Board, Issue 1679, pg. 130 – 138, 1999.

[24] Highet, R., "Optical Vehicle Tracking - A Framework And Tracking Solution", Honors Research Paper, Department of Computer Science, University of Canterbury, 2004.

[25] Ekinci, M., Gedikli, E., "Real-Time Background Model Initialization and Maintenance for Video Surveillance", International XII Turkish Symposium on Artificial Intelligence and Neural Networks- TAINN'2003, s. 1-10, Çanakkale, Turkey, 2003.

[26] Baldini, G., Campadelli, P., Cozzi, D., Lanzarotti, R., "A Simple And Robust Method For Moving Target Tracking", Proceedings of the IASTED International Conference Signal Processing, Pattern Recognition and Applications (SPPRA2002), pg. 108-112, 2002.

[27] Cheung, S-C. S., Kamath, C., "Robust techniques for background subtraction in urban traffic video", In Proceedings of the SPIE, volume 5308, pp. 881-892, 2004.

[28] Baker, M., Yanco, H. A., "A Vision-Based Tracking System for a Street-Crossing Robot", IEEE International Conference on Robotics and Automation, 2004.

[29] Visser, R., Sebe, N., Lew, M. S., "Detecting Automobiles and People for Semantic Video Retrieval," ICPR, p. 20733, 16th International Conference on Pattern Recognition (ICPR'02) - Volume 2, 2002.

[30] Zhang, C., Chen, S.-C., Shyu, M.-L., Peeta, S., "Adaptive background learning for vehicle detection and spatio-temporal tracking", Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia, Vol.2., pg.797 – 801, 2003.

[31] Di Stefano, L., Viarani, E., "Vehicle Detection And Tracking Using The Block Matching Algorithm", Proc. of 3rd IMACS/IEEE Int'l Multiconference on Circuits, Systems, Communications and Computer, pg. 4491 – 4496, 1999.

[32] Hubell, D., "Eye, Brain and Vision", Scientific American Library, 1988.

[33] Fei-Fei, L., Fergus, R., Perona, P., "Learning Generative Visual Models From Few Training Examples: An Incremental Bayesian Approach Tested On 101 Object Categories", IEEE CVPR Workshop of Generative Model Based Vision, 2004.

[34] Fergus, R., Perona, P., Zisserman, A.. "Object class recognition by unsupervised scale-invariant learning", Proc. CVPR, 2003.

[35] Forsyth, D. A., Haddon, J., Ioffe, S., "Finding objects by grouping primitives", Shape, Contour and Grouping in Computer Vision, Springer-Verlag, 2000.

[36] Lazebnik, S., Schmid, C., Ponce, J., "Semi-local affine parts for object recognition", In Proc. BMVC, pg. 959 – 968, 2004.

[37] Forsith, D. A., Ponce, J.. "Computer Vision–A Modern Approach", Prentice Hall, 2003.

[38] Riesenhuber, M., Poggio, T., "Models of object recognition", Nature Neuroscience, 3(Supp): pp 1199–1204, 2000.

[39] Heath, M.D., Sarkar, S., Sanocki, T., Bowyer, K.W., "A Robust Visual Method for Assessing the Relative Performance of Edge-Detection Algorithms", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, pg. 1338 - 1359, 1997.

[40] Ando, S., "Image Field Categorization and Edge/Corner Detection from Gradient Covariance", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, pg. 179 - 190, 2000.

[41] Canny, J.F., "Finding Edges and Lines in Images", Technical report no. 720, Massachusetts Institute of Technology, 1983.

[42] Canny, J.F., "A Computational Approach to Edge Detection", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, No.6, pg. 679 - 714,November 1986.

[43] Schneiderman, H., Kanade, T., "Object Detection Using the Statistics of Parts", International Journal of Computer Vision, 2002.

[44] Toyama, K., Krumm, J., Brumitt, B., Meyers, B., "WallFlower: Principles and practice of background maintenance", International Conference on Computer Vision, pp. 255-261, 1999.

[45] Masson, T., "CG 101: A Computer Graphics Industry Reference", Indianapolis, New Riders Publishing, 1999.

[46] Welch, G., Bishop, G., "An Introduction to the Kalman Filter." Department of Computer Science, University of North Carolina at Chapel Hill, 2004. (TR95-041)

[47] Yilmaz, A., Javed, O., Mubarak, S., "Object Tracking: A Survey", ACM Computing Surveys, Vol. 38, No. 4, Article 13, December 2006.

[48] Blake, A., Curwen, R., Zisserman, A., "A framework for spatio-temporal control in the tracking of visual contours", International Journal of Computer Vision, October 1993.

[49] Matteucci, P., Regazzoni, C.S., Foresti, G.L., "Real-time approach to 3D object tracking in complex scenes", Electronics Letters, Vol. 30, pgs 475-477, March 1994.

[50] Stark, K., Fuchs, S., "A method for tracking the pose of known 3D objects based on an Active Contour Model", Proceedings of the International Conference on Pattern Recognition, August 1996.

[51] Pollefeys, M., "Feature point extraction", July 12, 2000.

[52] Sharifi, M., Fathy, M., Mahmoudi, M. T., "A Classified and Comparative Study of Edge Detection Algorithms", International Conference on Information Technology, 2002.

[53] Roushdy, M., 'Comparative Study of Edge Detection Algorithms Applying on the Grayscale Noisy Image Using Morphological Filter", GVIP Journal, Vol. 6, Issue 4, December 2006.

[54] Saptharishi M., Hampshire II, J.B., Khosla, P., "Agent-based moving object correspondence using differential discriminative diagnosis", Proceedings of Computer Vision and Pattern Recognition, pp 652-658, 2000.

[55] Wixson, L., Selinger, A., "Classifying moving objects as rigid or non-rigid", Proc. DARPA Image Understanding Workshop, 1998.

[56] Kilger, M., "A shadow handler in a video-based real-time traffic monitoring system", In Proc. IEEE workshop on Applications of Comp. Vision, pp.11-18, 1992.

[57] Rosin, P.L., Ellis, T., "Image difference threshold strategies and shadow detection", In Proc. of the sixth British Machine Vision Conference, 1994.

[58] Kaew Tra Kul Pong, P., Bowden, R., "An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection", In Proc. 2nd European Workshop on Advanced Video Based Surveillance Systems, AVBS01. Sept 2001.

[59] Elgammal, A., Harwood, D., Davis, L., "Non-parametric Model for Background Subtraction", In Proc. of the 6th European Conference on Computer Vision-Part II, pp. 751-767, 2000.

[60] So, A.W.K.,. Wong, K-Y.K, Chung, R.H.Y.,. Chin, F.Y.L, "Shadow Detection For Vehicles By Locating The Object-Shadow Boundary", In Proc. IASTED Conference on Signal and Image Processing (SIP 2005), pp.315–319, 2005.

[61] Salvador, E., Cavallaro, A.,. Ebrahimi, T, "Shadow Identification and Classification using invariant Color Models", In IEEE Signal Processing Soc. Int'l Conf. Acoustics, Speech, and Signal Processing (ICASSP2001), IEEE Press, pp. 1545-1548, 2001.

[62] Horprasert, T., Harwood, D., Davis, L.S., "A Statistical Approach for Real-time Robust Background Subtraction and Shadow Detection", Proc. IEEE ICCV'99 Frame-Rate Workshop, Kerkyra, Greece, September 1999.

[63] Prati, A., Mikic, I.,. Trivedi, M.M,. Cucchiara, R, "Detecting Moving Shadows: Algorithms and Evaluation", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 25, No. 7, July 2003.

[64] Piccardi, M., "Background subtraction techniques: a review", IEEE International Conference on Systems, Man and Cybernetics, pp. 3099-3104, 2004.

[65] Stauffer, C., Grimson, W.E.L., "Adaptive background mixture models for real-time tracking", In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp 246-252, 1999.

[66] Butler, D., Bove,V.M. Jr., Sridharan, S., "Real-time adaptive foreground/background segmentation", In EURASIP Journal on Applied Signal Processing, pp. 2292-2304, 2005.

[67] McIvor, A. M., "Background subtraction techniques", In Proceedings of Image and Vision Computing, 2000.

[68] Lipton, A.J., Fujiyoshi, H., Patil, R. S., "Moving target classification and tracking from real-time video", Proceedings of IEEE Workshop on Applications of Computer Vision, 1998.

[69] Lipton, A. J., "Local Application of Optic Flow to Analyse Rigid versus Non-Rigid Motion", Technical Report CMU-RI-TR-99-13, Robotics Institute, Carnegie Mellon University, December, 1999.

[70] Barron, J., Fleet, D., Beauchemin, S., "Performance of optical flow techniques", International Journal of Computer Vision, 12 (1), pp42-77, 1994.

[71] Haritaoglu, I., Harwood, D., Davis, L.S., "W4: real-time surveillance of people and their activities", IEEE Trans. on Pattern Analysis and Machine Intelligence, pp 809-830, 22 (8), 2000.

[72] Porikli, F., Thornton, J., "Shadow Flow: A Recursive Method to Learn Moving Cast Shadows", IEEE International Conference on Computer Vision (ICCV), ISSN: 1550–5499, Vol. 1, pp. 891–898, 2005.

[73] Prati, A., Mikic, I., Grana, C., Trivedi, M.M., "Shadow Detection Algorithms for Traffic Flow Analysis: a Comparative Study", In Proc. IEEE Intelligent Transportation Systems, 2001.

[74] Sonka, M., Hlavac, V., Boyle, R., "Image Processing, Analysis, and Machine Vision", 2nd edition, Brooks/Cole Publishing Company, 1999.

[75] Poisson, S.-D., "*Recherches sur la probabilité des jugements en matières criminelles et matière civile*" ("Research on the Probability of Judgments in Criminal and Civil Matters"), 1838.

[76] Russ, J. C., "The image processing handbook", Boca Raton: CRC Press, 2002.

[77] Gonzalez, R. C., and Woods, R. E., "Digital Image Processing", Addison-Wesley Publishing Company, 1993.

[78] Chang, P., Krumm, J., "Object Recognition with Color Cooccurrence Histograms," IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Fort Collins, CO, 1999.

[79] Ross, S.M., "Introduction to Probability and Statistics for Engineers and Scientists", 3rd edition, Academic Press, 2004.

[80] Statewide Crash Statistics, Nevada Department of Transportation Safety Engineering Division, 2007.

[81] United Nations, Treaty Series, Volume 125, No. 1671, Part V, TRAFFIC LIGHT SIGNALS, Article 53, 1949.

[82] The MUTCD is published by the Federal Highway Administration (FHWA) under 23 Code of Federal Regulations (CFR), Part 655, Subpart F. http://mutcd.fhwa.dot.gov/pdfs/2003r1r2/pdf_index.htm

[83] Davies, E.R., "Machine Vision: Theory, Algorithms, Practicalities", Elsevier Inc., 3rd edition, 2005.

[84] Jiang, C., Ward, M.O., "Shadow Identification", Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp.606-612, 1992.

[85] Horn B. K. P., Schunck B. G., "Determining Optical Flow", Artificial Intelligence, pp. 185-203, 17(1981).

[86] Andrew Senior, Arun Hampapur, Ying-Li Tian, Lisa Brown, Sharath Pankanti and Ruud Bolle, "Appearance models for occlusion handling", Image and Vision Computing, Volume 24, Issue 11, 1 November 2006, Pages 1233-1243.

[87] Nevada Department of Motor Vehicles, "Nevada Driver's Handbook", October 2007, www.dmvnv.com.

[88] Serra, J., "Image Analysis and Mathematical Morphology", Academic Press, 1982.

[89] Faugeras, O., "Three-Dimensional Computer Vision: A Geometric Viewpoint", M.I.T. Press, 1993.

[90] Sezgin, M., Sankur, B., "Survey over image thresholding techniques and quantitative performance evaluation", Journal of Electronic Imaging, 13(1), pp. 146-165, 2004.

[91] Serrano, A., Conde, C., Rodrique-Aragon, L.J., Montes, R., Cabello, E, "Computer Vision Application: Real Time Smart Traffic Light", EUROCAST, Lecture Notes in Computer Science, Volume 3643, pp. 525-530, 2005.

[92] Jahne, B., Haubecker, H., "Computer Vision and Applications: A Guide for Students and Practitioners", Academic Press, 2000.

[93] Salim, F.D., Loke, S.W., Rakotonirainy, A., Krishnaswamy, S., "U&I Aware: A Framework Using Data Mining and Collision Detection to Increase Awareness for Intersection Users", 21st International Conference on Advanced Information Networking and Applications Workshops, 2007.

[94] Hu, W., Xiao, X., Xie, D., Tan, T., Maybank, S., "Traffic Accident Prediction Using 3-D Model-Based Vehicle Tracking", IEEE Transactions on Vehicular Technology, Volume 53, No. 3, May 2004.

[95] Wang, K-F., Jia, X., Tang, S., "A Survey of Vision-based Automatic Incident Detection Technology", IEEE International Conference on Vehicular Electronics and Safety, pp. 290-295, Oct. 2005.

[96] Atev, S., Arumugam, H., Masoud, O., Janardan, R., Papanikolopoulos, N.P., "A Vision-Based Approach to Collision Prediction at Traffic Intersections", IEEE Transactions on Intelligent Systems, Volume 6, No. 4, December 2005.

[97] Wong, C.Y., Qidwai, U., "Vehicle Collision Avoidance System", Sensors, 2004, Proceedings of IEEE, Volume 1, pp. 316-310, Oct. 2004.

[98] Automatic Crash Response. Retrieved February 23rd, 2008 from http://www.onstar.com/us_english/jsp/index.jsp

[99] Gorder, P.F., "Software Pinpoints Traffic Accident 'Hotspots'", The Ohio State University Research News, 2007. Retrieved February 23rd, 2008 from http://researchnews.osu.edu/archive/accident.htm

[100] Mohammed, M. I., Anupama, R., "Scene Adaptive Shadow Detection Algorithm", Proceedings of World Academy of Science, Engineering and Technology, Volume 2, January 2005.

[101] Zhao, M., Bu, J., Chen, C., "Robust background subtraction in HSV color space", Proceedings - SPIE The International Society For Optical Engineering, Issue 4861, pages 325-332, 2002.

[102] Sindhu, A.J., Morris, T., "Resolving Complex Occlusions of Objects during Tracking using Region based Segmentations", From Proceeding Signal Processing, Pattern Recognition, and Applications, 2006.

[103] Salas, J., Jimenez, H., Gonzalez, J., Hurtado, J., "Detecting Unusual Activities at Vehicular Intersections", IEEE International Conference on Robotics and Automation, April 2007.

[104] Bradski , G.R., "Computer vision face tracking for use in a perceptual user interface.", Intel Technology Journal, 2nd Quarter, 1998.

[105] Allen, J.G., Xu, R.Y.D., Jin, J.S., "Object Tracking Using CamShift Algorithm and Multiple Quantized Feature Spaces", Proceedings of the Pan-Sydney area workshop on Visual information processing, 2004.

[106] Kass, M., Witkin, A., Terzopoulos, D., "Snakes: Active contour Models", International Journal of Computer Vision, Volume (1) #4, pp. 321-331, 1988.

[107] Foley, J.D., van Dam, A., Feiner, S.K., Hughes, J.F., "Computer graphics principles and practice", 2nd edition, Addison-Wesley, 1990.

VITA

Graduate College
University of Nevada, Las Vegas

Dawn Marie Strianese

Home Address:
3641 Blairmoor Street
North Las Vegas, NV 89032

Degrees:
Bachelor of Science, Computer Science, 1994

Thesis Title: A Mathematical Model for Computerized Car Crash Detection using Computer Vision Techniques

Thesis Examination Committee:
Chairperson, Evangelos Yfantis, Ph. D.
Committee Member, Laxmi Gewali, Ph. D.
Committee Member, John Minor, Ph. D.
Graduate Facility Representative, Mohamed Kaseko, Ph. D.