DISCRETIZED CATEGORIZATION OF HIGH LEVEL TRAFFIC ACTIVITES IN TUNNELS USING ATTRIBUTE GRAMMARS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

DEMİRHAN BÜYÜKÖZCÜ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE
OF MASTER OF SCIENCE
IN
THE DEPARTMENT OF COGNITIVE SCIENCE

SEPTEMBER 2012

DISCRETIZED CATEGORIZATION OF HIGH LEVEL TRAFFIC ACTIVITES IN TUNNELS USING ATTRIBUTE GRAMMARS

Submitted by Demirhan Büyüközcü in partial fulfillment of the requirements for the degree of Master of Science in Cognitive Science, Middle East Technical University by,

Prof. Dr. Nazife Baykal _____
Director, Informatics Institute

Prof. Dr. Cem Bozşahin _____
Head of Department, Cognitive Science

Prof. Dr. Cem Bozşahin _____
Supervisor

Examining Committee Members:

Prof. Dr. Deniz Zeyrek _____
Cognitive Science, METU

Prof. Dr. Cem Bozşahin _____
Cognitive Science, METU

Assist. Prof. Dr. Cengiz Acartürk _____
Cognitive Science, METU

Assist. Prof. Dr. Perit Çakır _____
Cognitive Science, METU

Assist. Prof. Dr. Sinan Kalkan _____
Computer Engineering, METU

Date: September 14, 2012

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Demirhan Büyüközcü

Signature : _____

ABSTRACT

DISCRETIZED CATEGORIZATION OF HIGH LEVEL TRAFFIC ACTIVITES IN TUNNELS USING ATTRIBUTE GRAMMARS

Büyüközcü, Demirhan
M. Sc., Department of Cognitive Science
Supervisor: Prof. Dr. Cem Bozşahin

September 2012, 83 pages

This work focuses on a cognitive science inspired solution to an event detection problem in a video domain. The thesis raises the question whether video sequences that are taken in highway tunnels can be used to create meaningful data in terms of symbolic representation, and whether these symbolic representations can be used as sequences to be parsed by attribute grammars into abnormal and normal events. The main motivation of the research was to develop a novel algorithm that parses sequences of primitive events created by the image processing algorithms. The domain of the research is video detection and the special application purpose is for highway tunnels, which are critical places for abnormality detection. The method used is attribute grammars to parse the sequences. The symbolic sequences are created from a cascade of image processing algorithms such as; background subtracting, shadow reduction and object tracking. The system parses the sequences and creates alarms if a car stops, moves backwards, changes lanes, or if a person walks into the road or is in the vicinity when a car is moving along the road. These critical situations are detected using Earley's parser, and the system achieves real-

time performance while processing the video input. This approach substantially lowers the number of false alarms created by the lower level image processing algorithms by preserving the number of detected events at a maximum. The system also achieves a high compression rate from primitive events while keeping the lost information at minimum. The output of the algorithm is measured against SVM and observed to be performing better in terms of detection and false alarm performance.

Keywords: Event Detection, Earley's Parser, Attribute Grammar, Abnormality Detection, Traffic Analysis

ÖZ

TÜNELLERDEKİ ÜST SEVİYE TRAFİK AKTİVİTELERİNİN ÖZNİTELİK GRAMERİ SAYESİNDE AYRIK KATEGORİZASYONU

Büyüközcü, Demirhan
M. Sc., Bilişsel Bilimler Bölümü
Danışman: Prof. Dr. Cem Bozşahin

Eylül 2012, 83 Sayfa

Bu çalışma videoları kaynak alan olay algılama sistemleri üzerine bilişsel bilim bakış açısıyla bir çözüme odaklanmaktadır. Tez, otoyol tünellerinden alınan videoların sembolik simgeler yönünden anlamlı veriler oluşturup oluşturamayacağı ve oluşacak bu sembolik simgelerin öznitelik grameri tarafından normal ve anormal olaylar olarak ayrılabilecek dizilimlere ulaşıp ulaşamayacağı sorusunu sorar. Tezin ana motivasyonu, görüntü işleme algoritmaları tarafından oluşturulan ilkel sembolik olay dizilimlerini bölümlendirebilecek yenilikçi bir algoritma üretmektir. Araştırmanın ilgi alanı video algılama özel uygulama amacı ise anormallik tespiti için kritik yerler olan otoyol tünelleridir. Dizilimleri bölümlendirmek için kullanılan metod öznitelik grameridir. Sembolik dizilimler, arkaplan çıkarma, gölge azaltma, obje takibi gibi görüntü işleme algoritmaların ardışık çalışmasıyla oluşturulmaktadır. Sistem bu dizilimleri bölümlendirip araç durması, aracın ters yöne gitmesi, serit değiştirmesi, tünelde yaya yürümesi veya yayaların otoyola adım atmaları gibi olayları algılandığında alarmlar üretir. Sistem bu kritik durumlar Earley bölümlendiricisi sayesinde ayıklarken video işlerken gerçek zamanlı performansını korur. Tezin

yaklaşımı yakalanan olay sayısını maksimumda tutarken görüntü işleme algoritmalarından kaynaklanan sahte alarmları yüksek oranda filtreler. Sistem veri kaybını minimumda tutarken sembolik dizilimlerde yüksek bir sıkıştırma oranına ulaşmıştır. Algoritmanın çıktıları SVM algoritması ile karşılaştırılıp tespit performansı ve sahte alarm performansında daha yüksek başarıya ulaştığı gözlemlenmiştir.

Anahtar Kelimeler: Olay analizi, Trafik Analizi, Öznitelik Grameri, Anormallik Tespiti, Earley Bölümlendiricisi

ACKNOWLEDGEMENTS

I would like to thank my supervisor Prof. Dr. Cem Bozşahin for his abundant support, vision and guidance throughout the work involved in producing this thesis He inspired me by his approach to the problem and through challenging questions encouraged me to think critically about the topic.

My thanks also to Özcan Gülderen for helping me with the implementation and Başar Turgut for his invaluable help with the preparation of the dataset.

I offer my gratitude to my business partner Kaan Kayabalı for his patience through the whole process of preparing my thesis.

I am grateful to my family for their support without which this thesis would not have been possible. I also thank my friends; Emrah Bala, Türkü Eğinlioğlu and Barış Orhan for their enduring encouragement during my studies.

To My Family...

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVATIONS

BGS	Background Subtraction
HMM	Hidden Markov Model
ANPR	Automatic Number Plate Recognition
CCTV	Closed Circuit Television
PTZ	Pan-Tilt-Zoom
FOV	Field of View
CFG	Context Free Grammar
SCFG	Stochastic Context Free Grammar
AG	Attribute Grammar
OPENCV	Open Computer Vision Library
RANSAC	Random Sample Consensus
SVM	Support Vector Machine
CRF	Conditional Random Field

CHAPTER 1

INTRODUCTION

Recent developments in imaging, storage and computation of these images have led to a time where analysing real time video information is not only inexpensive but mandatory in a variety of domains. For over three decades, intelligent video analysis systems have been used to monitor various activities. Real time video processing capabilities have enhanced the quality of life in a number of different sectors and technology fields whether a typical user recognizes it or not in their normal life.

Typical applications that we have succeeded to automate include; people counting, traffic monitoring, factory product line monitoring, healthcare diagnostics, satellite imaging, mining and in almost all industries. Event detection systems have attracted great attention in the last decade since they promise intelligent systems that detect, compute and behave accordingly in any situation that could be useful for human life.

In the traffic domain, analyses of traffic violations, abnormal event detection, statistics collection and spotting certain drivers suspected of committing a crime have been the target on which the majority of products and research have focused. Traffic data collection and abnormal event detection have been the subjects that have been addressed by much research aiming to automate the process and minimize human errors. The main bottleneck of the current systems is the need for a vast amount of manpower to analyse the video streams continuously. Video analytics help to process this data in real time and continuously thus, overcoming this bottleneck. The number of cameras that need to be monitored can be in the thousands, and due to the 24-hour operation to the cameras can be vulnerable to errors.

It has been shown in various psychological studies that the detection capability of human operators watching hundreds of monitors declines over time (Gavin J.D. Smith, 1999, p377). The natural importance attributed to the task of abnormal event detection arises from the critical life-saving mission due to the early alarm generation capabilities that can prevent accidents. The increase in processing power has also led video analytic system creators to develop novel and more robust detection algorithms to deliver further life-saving functions.

Traffic monitoring systems have developed various skills in the past three decades. The general abilities of modern traffic analytics software include; ANPR (Automatic Number Plate Recognition), traffic violations (wrong turns, driving in the wrong direction, violation of speed limits, driving through a red light crossing etc.), event detections (stopped vehicle, disallowed pedestrian occurrence, critical low speed driving in highways and tunnels, incidence of traffic jams, wrong parking detection etc.). In these domains only ANPR technology uses cameras that are specifically zoomed to plates thus they can be only used for this purpose. For traffic violations and event detection purposes, a general wide-area observation camera and a computer vision software combination is adequate to meet the general detection criteria.

However, low resolution, low image quality, night-time and weather conditions that effect the iris and the decline in image quality degrade the overall operating performance. Another challenge arises from the resolutions that are used in surveillance cameras. A typical traffic surveillance system is composed of CCTV (Closed Circuit Television) cameras whereas for object detection tasks using a high resolution input is required to achieve a robust performance. The performance of a recognition task naturally increases with features given to the classifiers and features that are heavily dependent on the input resolution. This phenomenon is one of the problems that limit real-time operational success and challenges the current systems.

Figure1: Typical tunnel camera installation field of view (FOV)
The outlines show target Car Sizes calibrated into the FOV – Adana Bahçe Tunnels

Cameras used for event detection tasks are mainly stationary and mostly mounted on poles near the road or sideways in tunnels. PTZ (Pan Tilt Zoom) cameras can offer a similar resolution as stationary cameras but typical algorithms used for the analysis consist of moving object extraction techniques as pre-filters before recognition and further analytics tasks. Thus, background subtracting for moving object detection has become an industry standard since it lowers the computational complexity and the need for additional servers or an increase in the number of cameras/servers.

Focusing on the problem of event detection merely from a computer science perspective has brought event detection methods to maturity even though there are aspects that are not robustly working robustly in complex scenarios. In the event detection domain, where the majority of research is backed with computer science algorithms; new methods are inspired by cognitive science, AI and linguistics have been emerging in the last three decades (Moore & Essa, 2002, Ivanov & Bobbick, 2002) and these methods are basically biologically inspired. Symbolic representations of sub-events, abstractions of lower level primitive events, and

grammatical representations of sequential inputs have been some of the cognitively inspired perspectives that computer scientists have begun to borrow from cognitive scientists.

In this thesis a general system has been proposed and developed. The focus is on the event detection subsystem on a video analytics system designed to create real-time events occurring in a highway tunnel traffic scenario. An approach is developed to parse and recognize the string of symbols generated by the traditional computer vision systems, and to detect higher-level events in a highway tunnel scenario. The aim of using this grammatical parser is to increase the accuracy against complex and noisy outputs generated by the computer vision algorithms.

1.1 Motivation

In a typical tunnel surveillance scenario, there are critical events to be detected mainly for transportation security purposes. Tunnels have been the location of dramatic accidents in the past that resulted in serious casualties. Most of these accidents have been reported to be due to events that could have been avoided if events such as ‘Stopped Vehicle’, ‘Vehicle in wrong direction’, ‘Low speed traffic flow’, ‘Pedestrian walking’ had been detected instantaneously and action was taken by tunnel security staff.

The situation in tunnel surveillance differs from the typical outdoor traffic scene. Tunnels create a medium where the lighting does not vary as much as the outdoor scene which makes the detection and recognition tasks easier and more accurate than an outdoor setup.

Although the tunnel has controlled lighting conditions, this does not solve all the problems of event detection and there are still challenges for the robust detection of vehicles. The headlights of cars and the typical low resolution in the standard cameras (320 x 240 in most cases) create a problem for the background subtracting algorithms. These algorithms depend on the difference of values of pixels between the memorized and new images. Headlights and low resolution create an environment where non-motion pixels are labelled as motion pixels. The variety of vehicles (trucks, buses, cars, motorbikes etc.) takes the problem into more complex environments because the image characteristics for different types of vehicles are also different.

In short, there is a need for higher-level algorithms to disambiguate the complex and mostly faulty outputs of the computer vision algorithms. These algorithms can be higher-level structures such as machine learning or syntactic methods to deliver more robust detections with less failure. This thesis describes how input video is analysed with image processing algorithms to atomic primitive events, then how these primitive events are parsed and recognised through an attribute grammar to create accurate detection results in a domain, where precision means saving human lives.

1.2 Thesis Statement

This thesis defines a complete software system that delivers critical events defined in the tunnel scenario to be labelled automatically. The system comprises low-level vision algorithms used to create atomic symbols, and a grammatical approach to parse real world events from these atomic symbols. Computer vision algorithms that have been used for over a decade and are considered as standard in the domain are not within the scope of this thesis but these algorithms are summarised and explained. The main contribution of this thesis to the field is in the syntactic

definitions and parsing mechanism for a previously unexplored set of events in a previously unexplored video domain of tunnel traffic.

1.3 Thesis Overview

Chapter 2 focuses on the foundations on which the theory of the proposed method is based. Concepts, categorization, concept learning, abstraction, the force of ecology on cognition, affordances and events are covered. An overview is given of the types of grammars that can be used in a parser context.

Chapter 3 focuses on the general problem statement, the specific background of the theories, which are used through the thesis as attribute grammars. The Earley parser is also introduced.

Chapter 4 presents the material on the specific problem solution and implementation. The focus is on how the image processing layers and parser layers are designed and implemented as well as the challenges and options, which are faced through the implementation.

Chapter 5 discusses the experimental results that are implemented on the datasets. Information is given on the types of results that are observed and how these results are measured in detail.

Chapter 6 describes the importance of the findings. Then outlines the importance of the work that was undertaken and suggests how the proposed method can be further developed further in future.

CHAPTER 2

LITERATURE REVIEW AND BACKGROUND

To draw a conclusion in the domain of event categorization, it is necessary to understand the underlying mechanisms that inspire the categories of objects. To categorize objects, events or any internal or external entities that any organism face, the idea of concept should be understood first.

Concepts are mental entities, which represent the mind's intermediate information storage and representation blocks between layers (Machery, 2009). From this point of view, concepts need to be better studied to understand the similarity between the information processing mechanism between different units of the mind. The main link between concepts and the model in the thesis are the primitive events, which are generated by the image processing layer and consumed by the parsing layer.

Since in this thesis the main purpose is to categorize high level events, a comprehensive review of the categorization and event detection literature. This review mainly focuses on the categorization of objects; the categorization of events is also examined in terms of ecologic affordances.

Abstraction is also described in this chapter since in the thesis the event detection phenomena are approached by two abstracted layers; image processing and parsing. To understand why it is logical to dramatically separate these layers, the motive behind the abstraction concept has to be understood. Affordances and ecological approaches need to be covered because they affect how we perceive and categorize events depending on different contexts.

2.1 Concepts

Concepts, whether defined in psychological terms or philosophical terms, are the key elements in understanding how information processing and symbolic representation create an infrastructure between the layers of processing in the mind. Concepts are useful for us to understand how knowledge is represented, accessed or manipulated (Machery, 2009, p9). Whether an entity is solid for example; a 'cat' or vivid as 'feelings', the resulting representation in our minds is such that we can process both of them almost at the same complexity in our daily routine, whether the input is auditory, visual or lingual.

Solomon, Medin, and Lynch stress the importance of concepts claiming that concepts are the building blocks of thought. How concepts are formed, used and updated are, therefore, central questions in cognitive science (Solomon & Medin & Lynch, 1999).

Gelman and Medin propose:

Concepts function in enormously varied ways. Concepts can be used for extremely rapid identification (as when escaping from prey), organizing information efficiently in memory, problem-solving, analogizing drawing inductive inferences that extent knowledge beyond what is known, embodying and imparting ideological inferences, conveying aesthetic materials (e.g., metaphor, poetry), and so forth… In short conceptual functions go beyond categorization (Gelman & Medin 1993, p158-159)

Psychologists believe that a concept is a body of knowledge that is used by higher order cognitive functions for categorization, reasoning, learning and analogy creation (Machery, 2009). If we want a definition of concept without diving deeper into psychological debates, the most generalizable definition would be:

A concept of x is a body of knowledge about x that is stored in the long-term memory and that is used by default in the processes underlying most, if not all, higher cognitive competences when these processes result in judgements about x. (Machery , 2009, p12)

Concepts are sometimes also used interchangeably with the term 'mental representations' in psychology (Machery, 2009). Without investigating further the details of evidence for the existence of concepts or the arguments and definitions raised by philosophical approaches to concepts, three types of concepts that are widely accepted by cognitive scientists should be presented. These are; concepts as prototypes, exemplars and theories.

2.1.1 Concepts as Prototypes

Concepts as prototypes can be viewed under the heading of a probabilistic view or explanation-based views (Komatsu, 1992, Medin, 1989). This paradigm can also be called the knowledge approach (Murphy, 2002). The prototype paradigm of concepts is built around the idea that concepts are represented as prototypes. The prototype approach considers concepts as entities which store statistical knowledge about category members (Smith, 1989). The statistical knowledge is assumed to accumulate during the learning phase of a concept.

The attributes attached to a concept can be measurable quantities such as colour, salience, texture and proportions among various dimensions. Eventually what the approach suggests is that there exists only one modal for an individual concept. Any object that is to be matched against a concept is measured in terms of the mathematical similarity. How far the sample stands from the existing model is the basis for the final verdict (Hampton, 1993). The similarity measure Hampton proposes is;

$$S(x,C) = f(w(x,i))$$

where S(x,C) is the similarity between the target object x and the category C; f is a function mapping weights of values to similarity; and w(x,i) is the weight of the value possessed by x for the i^{th} attribute represented by the prototype.

2.1.2 Concepts as Exemplars

The paradigm of concepts as exemplars proposes that concepts are built around sample-like structures stored in the memory. These sample-like structures are called

sets of exemplars (Lee Brooks, 1978). Although the exemplars point of view resemble the prototype theory in terms of whether models are featural (Medin and Schaffer, 1978) or dimensional (Nosofsky, 1986), they make very different assumptions on how the long-term memory stores knowledge. Exemplar-based models assume that cognitive processes measure the similarity of a target with the exemplars stored in the mind whereas prototype theories measure the similarity of a target with the different parameters of a concept (Medin and Schaffer, 1978, 211-212, Nosofsky, 1986).

To create a higher similarity measure, a target has to be sufficiently similar to an exemplar in an exemplar concept model, but in a prototype concept the target has to be sufficiently similar to the statistical average of all the previous concepts that are involved in the parameters of the prototype (Medin & Schaffer, 1978). In this manner, the similarity measure of the exemplar is non-linear, however the similarity measure of the prototype concept can be linear or non-linear.

2.1.3 Concepts as Theories

Although there is an on-going debate two main trends exist among psychologists concerning the theory paradigm. The first group accepts concepts as theories (Rips 1995, Rehder 2003), whereas the other group considers them to be elements of theories (Gopnik & Meltzoff, 1997).

Theory theorists believe that causal, functional and nomological relationships are stored in the knowledge of concepts (Gopnik & Wellman, 1994). They think there is an underlying similarity between concepts and scientific theories in that they both obey systematic laws. These systematic laws correlate prediction with explanation.

Murphy and Medin emphasise the differences between concepts and scientific theories:

We use theory to mean any of a host of mental 'explanations,' rather than a complete, organized, scientific account. For example, causal knowledge certainly embodies a theory of certain phenomena; scripts may contain an implicit theory of the entailment relations between mundane events; knowledge of rules embodies a theory of the relations between rule constituents; and book-learned, scientific knowledge certainty contains theories. Although it may seem to be glorifying some of these cases to call them theories, the term denotes a complex set of relations between concepts, usually with a causal basis. Furthermore, these examples are similar to theories used in scientific explanation (Murphy & Medin 1985, p290).

A straightforward example can be given to clarify the causality in concepts:

If at a party, a guest jumps into the swimming pool with her clothes on, we can conclude that she is drunk. This categorization judgement does not result from matching the concept of drunken people with a representation of this guest. On the contrary, we infer that the most plausible explanation for the behaviour of this guest is that she is drunk (Murphy & Medin 1985, p295).

2.2 Categorization

Our minds are bombarded with information about the concepts that are all around us (Mervis & Rosch, 1981). If we cannot generalize our knowledge about categories or classes, we would only be left with individual concepts, which would be an unmanageable amount of individual information.

Even though there are debates on what categorisation is, we can define it as the ability to judge whether an individual concept belongs to a certain class or not (Hampton & Dubois, 1993). Psychologists believe that agreeing that a concept has a membership relationship with a class helps us to extend the information that can be processed about that concept furthermore, to the perceptible input inferred from the concept at that time.

Inclusion judgements are the kind of reasoning in which a cognitive unit decides whether a class is included in another class (Machery, 2009). Many psychologists believe that, these decisions are produced by a single cognitive process for both types of judgements. Our perceptual system or linguistic system creates the input that is connected to our categorization processes.

The motivation of categorization process varies. We may categorize an object to learn it and enter it into our knowledge base or to make a decision during an action (Thorpe, Deloerme & VanRullen 2001). The resulting confidence level of categorization also varies with the motivation even if the input is the same.

2.2.1 Concept Learning

Concept learning is widely defined, as the capacity to acquire concepts (Fodor, 1981). It can be exemplified as learning about a fruit by being told about it, however, if you acquire the concept of a new fruit when travelling abroad, this process is narrowly defined concept learning (Machery, 2009).

Clark Hull attempts to describe how a child typically learns a new concept:

A young child finds himself in certain situation…and hears it called 'dog.' After an indeterminate intervening period he finds himself in a somewhat different situation,

and hears that called 'dog.'... Thus, the process continues. The 'dog' experiences appear at irregular intervals. The appearances are thus unanticipated. They appear with no obvious label as to their essential nature. This precipitates at each new appearance a more or less acute problem as to the proper reaction…. Meanwhile the intervals between the 'dog' experiences are filled with all sorts of other absorbing experiences, which are contributing to the formation of other concepts. At length the time arrives when the child has a 'meaning' for the word dog. Upon examination this meaning is found to be cats, dolls and teddy bears. But to the child the process of arriving at this meaning or concept has been largely unconscious (Hull, 1920, 5-6).

Throughout the 20th century there have been a wide range of research undertaken and experiments carried out on concept learning and categorization. In this research groups of subjects are presented with types of concepts before the actual dataset is presented to them, for the purpose of training (Machery, 2009). In these experiments, the main motivation was to detect the methods, success rates, decision durations and a general model of categorization cognition. In some of these experiments subjects were exposed to a single class of objects whereas in others they were presented with multi classes of objects. In the former experiments, the participants were mostly given a set of negative samples as well as the positive samples. The measurement of learning is based on the success of correctly categorising a fixed number of items or due to performance on achieving the task within a specific time frame. The most common factor for evaluating learning is widely accepted to be the rate of learning.

Concept acquisition can be performed supervised or unsupervised and can occur, as Hull suggests, by encountering different members of the same category over time (Hull, 1920, 5-6). Children can learn concepts and categories via both unsupervised and supervised situations. For example, a parent may present a concept to their child by drawing the boundaries of an object or delivering the information in a purely lingual manner.

Learning concepts and categories may differ according to age, vary from person to person and from concept to concept. A category can also be learned by merely reading the definition in a dictionary (Machery, 2009). There are some experiments that suggest children can learn some concepts at their first exposure to the concept (Carey and Bartlett, 1978) whereas other concepts such as in the field of mathematics may take years to acquire. As a result, irrespective of the underlying context representation mechanisms, it is widely believed that the cognitive process for learning a category with encountering new members is a single process (Nosofsky, Zaki, 1998; Nosoftsky & Johansen, 2000).

2.3 Abstraction

Herbert A. Simon stated the importance of abstraction as follows:

Scientific knowledge is organized in levels, not because reduction in principle is impossible, but because nature is organized in levels, and the pattern at each level is most clearly discerned by abstracting from the detail of the levels far below. …. And nature is organized in levels because hierarchic structures - systems of Chinese boxes- provide the most viable form for any system of even moderate complexity (Herbert A. Simon, 1973, 1-28).

How the mind manages to model and process the complexity of the world is one of the core questions of cognitive science. To understand this we must accept the fact that the vast computational and problem solving processes of the mind are created from a series of simple and exact stages carried out in different layers.

Hierarchical abstraction presents how these simple stages, even if they are perfectly simple in their own contexts, form an incredible amount of complexity (Edelman,

2008). It would be hard to explain abstraction better than as in the words of Robert Penn Warren:

Simplicity is what complexity must be made of, because there isn't anything else to make it out of, and hierarchical abstraction is the only way in which sufficiently interesting complex stuff can be built out of simple building blocks (Edelman, 2008, p30).

David Marr gives a good example of how roman numbers and modern numbers differ based on the parity of a number, in his famous book *Vision* (Marr, 1982). In roman notation it is impossible to tell if a number is odd or even by looking at the number whereas in modern notation we can easily conclude by just looking at the rightmost digit. Imagine if one of the most important attributes for a number is whether it is a prime number. We could easily represent this kind of concept by adding a parity bit to the number, if we really need that kind of representation. In short, the representation we choose for an entity can make it simpler or more complex to process.

Edelman claims that:

Hierarchical representation is not a tool that cognitive scientists use to understand the mind, but it is the mind's tool for understanding the world (Edelman, 2008, p31).

A complex system such as the mind can only be understood if we divide its total reasoning into meaningful blocks of simpler representations and functions. If the simpler blocks are not organized in such a manner that the information can travel and transfer between blocks, we would not understand the working of a cognitive system that is able to deal with a complex world (Marr and Poggio 1977). Thus, hierarchical representations help the mind abstract some kind of information from some blocks and scale up the total understanding and reasoning of the overall system.

2.4 Affordances and Ecological Effects of Medium

Affordance is what the medium offers to animals or what it provides that assists the animal in achieving something. Affordances arise from the fact that both humans and animals see any animate or inanimate in terms of the benefits that can be gained by using it (Gibson 1966).

From the point of view of affordance, objects are not to be categorized by what they are, but how they are to be understood and used in different contexts. For affordances to be meaningful, the size of the object should be considered close to the animal's size. The earth, a canyon or a mountain holds no (or minimal) affordances for an animal that is of human size. A tree, a cave, a sharp object has the affordances of eating, hiding and using as a cutting device respectively. Terrestrial surfaces can be used for climbing, objects like a stool can be used for sitting or a flat surface can be used for lying upon or standing on. Same objects can afford different activities for different animals (Gibson, 1986).

Ecologists refer to the concept of *niche*, which is a space or medium that a certain animal occupies. However, this occupancy is not referred to as a habitat but as a set of affordances offered by the environment to the animal. The environment affords various activities to the animals such as; many kinds of nutrients on which the animal can feed, caves for the animal to hide in, various kinds of materials to make tools, shelters; and a wide range of terrains that make it possible for the animal to swim, crawl, run, walk, climb (Gibson, 1986).

Concerning the environment Gibson argues that:

An important fact about the affordances of the environment is that they are in a sense objective, real, and physical, unlike values and meanings, which are often supposed

*to be subjective, phenomenal, and mental. But, actually, an affordance is neither an objective property nor a subjective property; or it is both if you like. An affordance cuts across the dichotomy of subjective-objective and helps us to understand its inadequacy. It is equally a fact of the environment and a fact of behaviour. It is both physical and psychical, yet neither. An affordance points both ways, to the environment and to the observer… The organism depends on its environment for its life, but the environment does not depend on the organism for its existence (*Gibson *1986*, p129).

The chemical, physical, meteorological and geographical conditions of the environment are what make it possible for animal to live on earth. Air affords breathing, a flat surface affords being laid or standing upon; flint, clay or other deformable entities afford to be shaped and staying strong afterwards. As a horizontal substance affords walking, a vertical substance affords climbing on. As civilization flourished we invented steps for vertical spaces to allow us to ascend them. Apart from the objects that are attached to the earth, the objects that we call detached are also of various kinds and have affordances.

2.4.1 Affordances of Detached Objects:

a) Elongated objects of moderate sizes afford hitting or striking such as a club or hammer. These kinds of objects can also be used for levering heavier objects. If they are tiny they can be used as needles, or if they are sharp but large, they can be used as spears.

b) A solid object with adequate size, sharpness and strength can be used as a knife for cutting or attacking.

c) An object which is graspable can be used as a fun element such as a ball or a stone or missile to be thrown at other animals. These missile objects can be combined with bows or catapults to attack larger groups of species or even larger buildings such as castles. The ability to use objects as missiles makes human a very dangerous species with respect to the rest of the animal kingdom.

d) An elastic object, which can be elongated, can be used as a rope, can function as a manufacturing for higher units of equipment.

e) Lastly, a hand-held object, which can mark caves, trees or any affordable surface is an extremely useful device. It can be a brush, pen or pencil as long as it marks a surface, it helps people to write, create symbols words and convey linguistic meaning on surfaces (Gibson, *1986*, p133).

A stone thrown directly at an animal can be a life-threatening hazard whereas a stone can also be used as a paperweight or a hammer, it can also be used with other stones to build a wall. So theory of affordances helps us to think about classical class categorization. There are no clear-cut definitions of objects in affordances. Perception is economic and we do not have to classify every feature or detailed class information of an object to realize what it affords (Gibson 1966, p286).

Affordances are exerted on an animal by surfaces, substances, places or other animals. These affordances are important because they can offer opportunity give harm, injury or even result in the death of the animal. Affordances are neither solely physical nor phenomenal because they take their reference from the observer, not only from the afforded object. (Gibson, *1986*, p143)

2.4.2 Event Detection as an Ecologic Phenomena

The definition of an event is different in different disciplines. However, in the context of this thesis some properties of events can be defined as:

- Taking a period of time
- Built of smaller semantic unit building blocks
- Using the salient aspects of the recognition phenomena (Lavee & Rivlin & Rudzsky 2009)

The optical information we perceive changes over time. This change created by disturbances in optical array information can result from various ecological phenomena. These disturbances alone cannot be linked directly with events because the motions of spots cause these optical disturbances and the cause behind the spots cannot be solely linked to the motions of objects (Gibson, 1968b). Events may be categorized as a change in; the layout of surfaces, color and texture of surfaces, and in the existence of surfaces. Gibson summarises the causes of these changes as:

The cause of change in layout of surfaces is forces; the cause of change in color is the change in the composition of the substance; and change in the existence of a surface is caused by a change in the state of the substance (Gibson, 1986, p94).

To understand which of these categories fit better in a tunnel domain event detection problem, further details of these categories are given in the following sections.

2.4.2.1 Change of Layout due to Complex Forces

This kind of change refers to any kind of alteration in the shape of the environment. These changes can occur by translational or rotational movements of objects, spinning, falling, turning, colliding, bouncing back, inanimate changes like a drop of fluid falling due to gravity, animate changes such as the posture changes of an animal, waves, flow, elastic changes, cracking, disintegration and explosions *(Gibson, 1986*, p95).

At this level of analysis the world is assumed to be stationary, and the objects are assumed to be changing places frame by frame *(Gibson, 1986*, p96). The interesting aspect of these kinds of events is that they can occur in combination with each other concurrently; an inanimate object falling to the ground, a collision, bouncing back and then another train of events. Mankind has invented a great number of machines and mechanical parts by mastering their characteristics of motion. The wheel, roller, crank, lever, pendulum, piston and motors are examples of this kind of machinery.

There is another interesting angle to these kinds of changes. The reversibility of an action is sometimes possible, and sometimes not. Displacements, translations, rotations and any kind of locomotion have a reverse movement opposing the original movement, so they can be reversed *(Gibson, 1986*, p96). Whereas breaking up, destroying, blowing up and disintegration has no reversibility.

2.4.2.2 Change of Color and Texture due to Changes in Composition

These kinds of changes may refer in plants to greening (increase in chlorophyll), fading (decrease in chlorophyll), ripening (increase in sugar), and flowering

(presence of nectar); for animals changes such as; coloration of skin (may refer to sexual receptivity), change of plumage (maturity), change of fur (onset of winter) on animal surfaces; and weathering of rock (oxidation), blackening of wood (fire), reddening of iron (rusting) can be seen on terrestrial surfaces *(Gibson, 1986*, p98).

When there is a chemical reaction it changes the substance in a way that it is irreversible and this event changes the color and texture of the entity. Animals should understand this change by visual cues before making a contact with the object to prevent any chemical damage (Gibson 1966b, Ch. 8).

Some kind of significant surface changes are correlated by multiple clues. Leaves change color as winter approaches. The flames of a fire can be considered more of an ecological event but it is accompanied with deformations and the motion of flames, which can also be considered a chemical reaction. It also ends with objects disappearing or mostly changing shape due to having been burnt. In animals fire can be detected by nose, skin, ears as well as eyes (*Gibson, 1986*, p98).

2.4.2.3 Change of Surface Existence

In this context when ice or snow melts the surface is changed radically so it is perceived as destroyed. When water evaporates the surface of water vanishes so it is also observed as destroyed. Similar events can be listed as the change from liquid to gas (evaporation, boiling), solid to gas (sublimation), cloud to gas (dissipation), solid to liquid (melting), solid to liquid (melting), solid into solution (dissolving), gas to liquid (condensation, rain), gas to solid, gas to cloud (formation), liquid to solid (freezing), solution into solid (crystallization, precipitation), disintegration, biological decay, destruction, aggregation, biological growth and construction. (*Gibson, 1986*, p 99). When an organism dies the surface disintegrates, so the surface

is destroyed However, ecological surface creation is not so easy to observe. An example can be seen as the growth of animals and plants.
As can be clearly seen the changes in water and ice are reversible but those related to organisms are not reversible. To conclude the subject of events by summarising some distinct properties:

Recurrence and Nonrecurrence: Recurrence and nonrecurrence always exist in the nature of ecological events. Events repeat whereas events may be considered as unique. Gibson quotes: *'Each new sunrise is like the previous one and yet unlike it, and so is each new day. An organism, similarly, is never quite the same as it was before, although it has rhythms.' (Gibson, 1986, p101).*

Reversible and Nonreversible Events: Some events are considered to be reversible and some non-reversible. As mentioned above, a change of position can be reversed but longer events that consist of shorter events cannot easily be turned back (Gibson, 1986, p101).

The Nesting of Events: Natural ecological events consist of units that are nested within each other. If a unit is not decided by the observer, the depth and nestedness and number of episodes cannot be counted (Gibson, 1986, p101).

2.5 Grammars

The event recognition problem in the tunnel scenario can be defined as a syntactic pattern recognition problem using grammars. The grammars used in language define the rules by which simpler constituents can form larger ones. The same model for language can be used to define activities that are spread over time and are built from simpler activity primitives. When the rules for the formation of a sentence or a complex activity are defined, using an appropriate method for parsing can recognize

a sentence or an event. The reason syntactic methods became popular in the video processing domain is that they are successful when the nature of a process is near random or very complex making it difficult to learn, and where the structure may be known a priori.

Using grammars in visual recognition is inspired by the work of Fodor who believes that language can be the ultimate formation in the mind to represent any phenomena in the world. The nested structure and the forms creating new forms can be the answer to modelling and computing how the real world is represented in the mind (Fodor, 1975). This is why grammar can be the way to represent the visual event recognition phenomena.

2.5.1 Deterministic Context Free Grammars

One of the earliest use of grammars for visual event detection was proposed by Brand, who uses a grammar to detect and recognize manipulations made on objects by human. The grammar is enforced by natural assumptions such as objects cannot leave the scene without human intervention and objects cannot be manipulated without human intervention (Brand et Al., 1996). These assumptions are rooted on the principles of causality of motion and are defined by Brand as:

- The principle of *contact* implies that there can be no action at a distance from an object and there can be no contact without action.
- The principle of *cohesion* implies that there can be no fusing or splitting unless a combination of causal events creates forces that compel the objects to fuse or split. This rule guarantees if there is no causal reason the identities of individual objects remain unchanged over time.

- The principle of *continuity* implies every object must have a stable trajectory without connected dots and no two objects can occupy the same space in same time without a contact relation.
- The principle of *animacy* implies there can be no contactless acceleration without agency or gravity.

On the basis of these assumptions, the video of a person fixing a computer is used as dataset and an attempt is made to recognise events by attaching subevents as enter, remove, detach, and exit to the foreground blobs in the video frame. These subevents then are searched for alternative possible parses and the best fit for the string is selected.

Figure2: Samples of dataset from Brand et al., (1996)

Other researchers have approached the problem of recognising human activities in a different manner. They have used visual detectors to identify the position and velocity of the head, upper body and lower body (Ryoo & Aggarwal, 2006). The position and velocity information of these visual detectors are then turned into poses and gestures by Hidden Markov Models (HMM), which is a popular recognizer if the input of the system is fairly invariant in terms of feature outputs. HMM outputs are then searched and parsed using changing time intervals to detect different interactions between people.

The system created by Ryoo and Aggarwal (2006) can detect composite actions, which are composed of atomic actions from the same action owner or two different action owners. The authors defined a list of primitive events to be detected and a set of production rules defining the higher level activities of interest. They parsed the gesture information in terms of spatial, temporal and logical constraints. The spatial information concerns whether the objects are too close to perform an interaction and the temporal information comprises the sequential occurrence of subatomic events that add up to a complex event.

Logical constraints such as ‘and’, ’not’, ’or’ are used to combine whether events occur together or independent of each other. The use of CFGs by Ryoo and Aggarwal is not stochastic but deterministic.

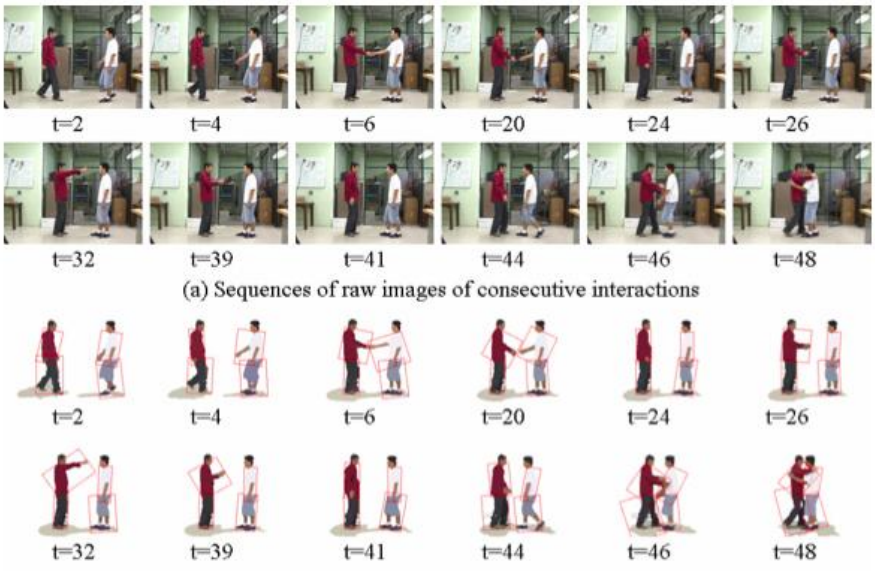
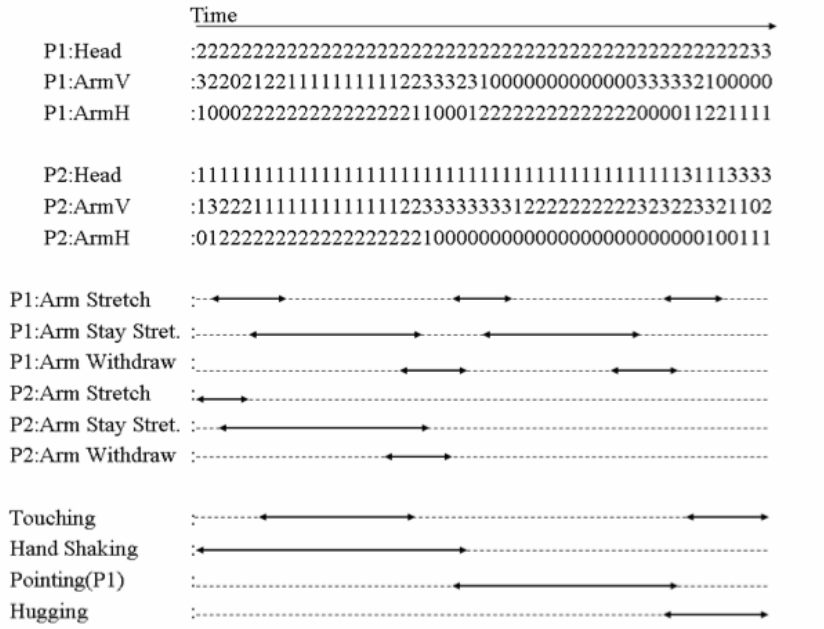


Figure 3: Sequence of actions for the approach devised by Ryoo & Aggarwal

Table 1: Temporal actions table for the approach devised by Ryoo & Aggarwal



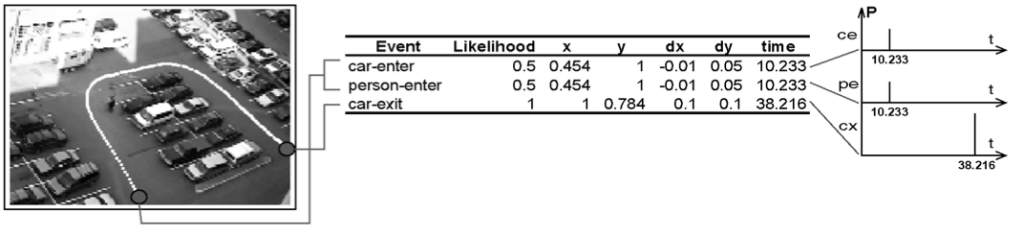
Deterministic grammars are prone to detection errors since they rely heavily on the performances of a low-level detector (Turaga et. al. 2008). Tracking errors or other kinds of observation errors in low-level detectors result in poor performance of the CFGs used in higher levels. Formulating the grammatical rules for multi-object interactions, parallel events and overlapping events is another challenge for deterministic grammars since the complexity of rules expands infinitely with the complexity of the possible scenarios.

2.5.2 Stochastic Context Free Grammars

Ivanov and Bobbick (2000) attempted to solve the activity recognition problem using of Stochastic Context Free Grammars (SCFGs). Their approach was mainly to divide the recognition problem into two layers; low-level detection and high level activity

parsing. They assumed that the structure of semantic activities is known so the input stream in the context should fit the pre-defined model.

Table2: Probabilistic grammar structure from Ivanov and Bobbick, (2002)



For the low-level detectors they used HMM to generate discrete symbols. These *symbols* are then fed into the SCFGs that are used to parse the stream. The importance of the work of Ivanov and Bobbick lies in the fact that they included algorithms for insertion, deletion and skipping the stream to handle ambiguity. The skip transition is a concept they followed in grammar production rules and it resulted in an increased robustness in overall system performance.

Table 3: Parsing Rules used by Ivanov and Bobbick

G_p :			
TRACK	→	CAR-TRACK	[0.5]
	|	PERSON-TRACK	[0.5]
CAR-TRACK	→	CAR-THROUGH	[0.25]
	|	CAR-PICKUP	[0.25]
	|	CAR-OUT	[0.25]
	|	CAR-DROP	[0.25]
CAR-PICKUP	→	ENTER-CAR-B CAR-STOP PERSON-LOST B-CAR-EXIT	[1.0]
ENTER-CAR-B	→	CAR-ENTER	[0.5]
	|	CAR-ENTER CAR-HIDDEN	[0.5]
CAR-HIDDEN	→	CAR-LOST CAR-FOUND	[0.5]
	|	CAR-LOST CAR-FOUND CAR-HIDDEN	[0.5]
B-CAR-EXIT	→	CAR-EXIT	[0.5]
	|	CAR-HIDDEN CAR-EXIT	[0.5]
CAR-EXIT	→	car-exit	[0.7]
	|	SKIP car-exit	[0.3]
CAR-LOST	→	car-lost	[0.7]
	|	SKIP car-lost	[0.3]
CAR-STOP	→	car-stop	[0.7]
	|	SKIP car-stop	[0.3]
PERSON-LOST	→	person-lost	[0.7]
	|	SKIP person-lost	[0.3]

They also implemented a real-time running system which can consistently check the output generated by the parser. They tested their algorithm on three different setups; hand gesture recognition, musical conductor recognition and person drop-off detection in a parking lot.

Table 4: Probabilistic Parser Table for the Ivanov and Bobbick Parking Lot scenario

DROP-OFF

Event	UID	Avg. Size	Class	P	x	y	t	frame
ENTER	724	0.122553	0	0.5	0.450094	0.938069	917907137.8	1906
ENTER	665	0.046437	1	0.5	0.6107	0.94674	917907122.5	1799
PERSON-LEAVE	665	0.045869	1	0.997846	0.648089	0.98855	917907142.7	1938
STOPPED	724		0	0.995784	0.348569	0.345513	917907146.5	1964
ENTER	780	0.034293	1	0.5	0.74188	0.980292	917907151.3	1998
ENTER	790	0.069093	0	0.5	0.814565	0.032611	917907153.4	2012
FOUND	787	0.033573	1	0.5	0.297585	0.357887	917907153.1	2010
CAR-LEAVE	790	0.061263	0	0.997285	0.975971	0.211984	917907155.3	2025
PERSON-LEAVE	780	0.038616	1	0.999923	0.974494	0.865237	917907158.6	2047
PERSON-LEAVE	787	0.032045	1	0.999997	0.296519	0.183704	917907158.7	2048
ENTER	813	0.034776	1	0.5	0.012821	0.348379	917907160.9	2063
ENTER	816	0.093513	0	0.5	0.960425	0.793899	917907161.9	2070
CAR-LEAVE	724	0.097374	0	0.993211	0.972272	0.693728	917907165.2	2091
CAR-LEAVE	816	0.089424	0	0.99023	0.693699	0.990798	917907165.2	2091

DRIVE-IN

In another study, SCFGs are used to detect interactions created by multi agents in a Blackjack game (Moore et al., 2002) applying a two level approach as used by Ivanov and Bobbick Their first detection layer uses visual hand detectors and trackers, powered by template matching algorithms. The outputs of these first level

detectors are fed to a SCFG to detect higher-level events such as who is winning the game or whether the player is a professional or novice. Their SFCG uses scanning, insertion and deletion approaches followed by Viterbi algorithm to detect the meaning in the game. Since the game has definite rules and contains a relatively small lexicon of primitive events their approach is well suited to the domain and is thus successful. It is possible to track two hands simultaneously through the game scene a multi-agent system can be developed.

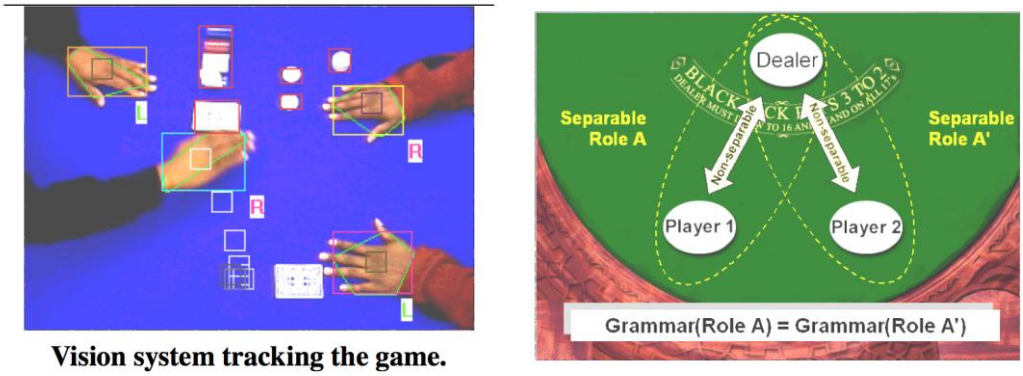


Figure 4: Event detector system in the blackjack game from Moore et al. (2002)

Table 5: 'Production rules for blackjack game from Moore et al. (2002)

Production Rules			Description
S	$\rightarrow AB$	[1.0]	Blackjack → "play game" "determine winner"
A	$\rightarrow CD$	[1.0]	play game → "setup game" "implement strategy"
B	$\rightarrow EF$	[1.0]	determine winner → "eval. strategy" "cleanup"
C	$\rightarrow HI$	[1.0]	setup game → "place bets" "deal card pairs"
D	$\rightarrow GK$	[1.0]	implement strategy → "player strategy"
E	$\rightarrow LKM$	[0.6]	eval. strategy → "dealer down-card" "dealer hits" "player down-card"
	$\rightarrow LM$	[0.4]	eval. strategy → "dealer down-card" "player down-card"
F	$\rightarrow NO$	[0.5]	cleanup → "settle bet" "recover card"
	$\rightarrow ON$	[0.5]	→ "recover card" "settle bet"
G	$\rightarrow J$	[0.8]	player strategy → "Basic Strategy"
	$\rightarrow Hf$	[0.1]	→ "Splitting Pair"
	$\rightarrow bfffH$	[0.1]	→ "Doubling Down"
H	$\rightarrow l$	[0.5]	place bets
	$\rightarrow lH$	[0.5]	
I	$\rightarrow ffI$	[0.5]	deal card pairs
	$\rightarrow ee$	[0.5]	
J	$\rightarrow f$	[0.8]	Basic strategy
	$\rightarrow fJ$	[0.2]	
K	$\rightarrow e$	[0.6]	house hits
	$\rightarrow eK$	[0.4]	
L	$\rightarrow ae$	[1.0]	Dealer downcard
M	$\rightarrow dh$	[1.0]	Player downcard
N	$\rightarrow k$	[0.16]	settle bet
	$\rightarrow kN$	[0.16]	
	$\rightarrow j$	[0.16]	
	$\rightarrow jN$	[0.16]	
	$\rightarrow i$	[0.18]	
	$\rightarrow iN$	[0.18]	
O	$\rightarrow a$	[0.25]	recover card
	$\rightarrow aO$	[0.25]	
	$\rightarrow b$	[0.25]	
	$\rightarrow bO$	[0.25]	

Symbol	Domain-Specific Events (Terminals)
a	dealer removed card from house
b	dealer removed card from player
c	player removed card from house
d	player removed card from player
e	dealer added card to house
f	dealer dealt card to player
g	player added card to house
h	player added card to player
i	dealer removed chip
j	player removed chip
k	dealer pays player chip
l	player bets chip

2.3 Attribute Grammars

Attribute grammars, which are explained in more detail in the following chapter, attach attributes to the symbols of a formal grammar. These attributes arise because symbolic representations cannot represent all the information in a symbol. If all the features of an entity are embedded in the symbolic representation, the representation will be so complex that it will degrade the performance of the overall system. Joe and Chapella (2006) used probabilistic attribute grammars in their work to represent a parking lot scene. They used primitive events as 'car_appeared', 'car_disappeared', 'person_appeared', 'person_disappeared' with attributes of location of objects, class of objects and the identity of the objects. With the help of these attributes and primitive events they were able to recognize the events in the parking lot as someone being picked up or someone getting in the car and driving off (Joe & Chapella 2006). The detected events can be seen in Figure 5 and the related grammar is given in Table 6.

Table 6: Attribute Grammar for normal events for research undertaken by Joe and Chapella (2006)

Grammar productions	Attribute rules and Semantic conditions
$PARKINGLOT \rightarrow PARKING_N \mid PARKOUT_N \mid DROPOFF_N$	
$PARKINGLOT \rightarrow PICKUP_N \mid WALKTHRU_N \mid CARTHRU_N$	
$PARKING \rightarrow CARPARK_0 perapp_N disappear_2 carstat_1$	(Near(X2.loc,X1.loc), sNearPt(X3.loc, BldgEnt))
$PARKING \rightarrow CARPARK_0 perapp_N carstat_1 disappear_2$	(Near(X2.loc,X1.loc), sNearPt(X4.loc, BldgEnt))
$CARPARK \rightarrow carapp_0 carstart_1 carstop_1$	X0.loc := X3.loc (NotInside(X1.loc,Fov), sInside(X3.loc, PkSpace1, PkSpace2))
$CARSTOP \rightarrow carstop_0 carstart_1 CARSTOP_1$	X0.loc := X3.loc
$CARSTOP \rightarrow carstop_0$	X0.loc := X1.loc
$PARKOUT \rightarrow perapp_0 disappear_1 carapp_N CARSTART_3 disappear_3$	(sNearPt(X1.loc,BldgEnt),Near(X3.loc,X2.loc), NotInside(X5.loc,Fov))
$CARSTART \rightarrow carstart_0 carstop_1 CARSTART_1$	X0.loc := X1.loc
$CARSTART \rightarrow carstart_0 carstop_1$	X0.loc := X1.loc
$CARSTART \rightarrow carstart_0$	X0.loc := X1.loc
$DROPOFF \rightarrow CARSTAND_0 perapp_N disappear_2 CARSTART_1$	(Near(X2.loc,X1.loc), sNearPt(X3.loc,BldgEnt))
$DROPOFF \rightarrow CARSTAND_0 perapp_N CARSTART_1 disappear_2$	(Near(X2.loc,X1.loc), sNearPt(X4.loc,BldgEnt))
$CARSTAND \rightarrow carapp_0 carstart_1 CARSTOP_1$	X0.loc := X3.loc (NotInside(X1.loc,Fov))
$PICKUP \rightarrow perapp_0 disappear_1 CARSTART_N disappear_3$	(sNearPt(X1.loc, BldgEnt), Near(X3.loc,X2.loc), NotInside(X4.loc,Fov))
$WALKTHRU \rightarrow perapp_0 disappear_1$	(NotInside(X1.loc,Fov), NotInside(X2.loc,Fov), sFar(X2.loc,X1.loc))
$CARTHRU \rightarrow carapp_0 CARSTART_1 disappear_1$	

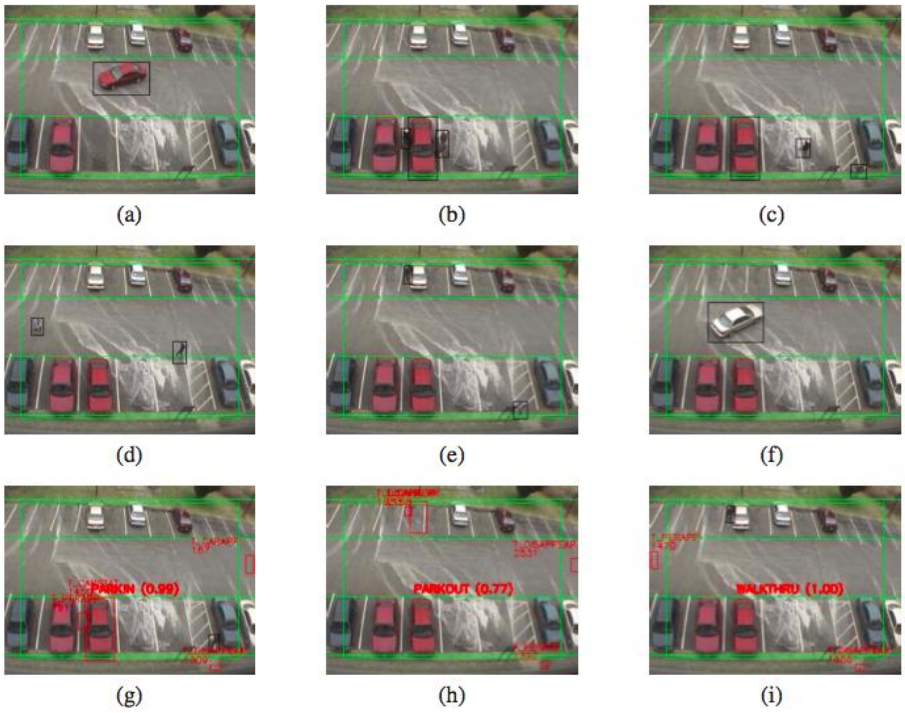


Figure 5: Parking scene for research undertaken by Joe and Chapella (2006)

CHAPTER 3

PROBLEM STATEMENT

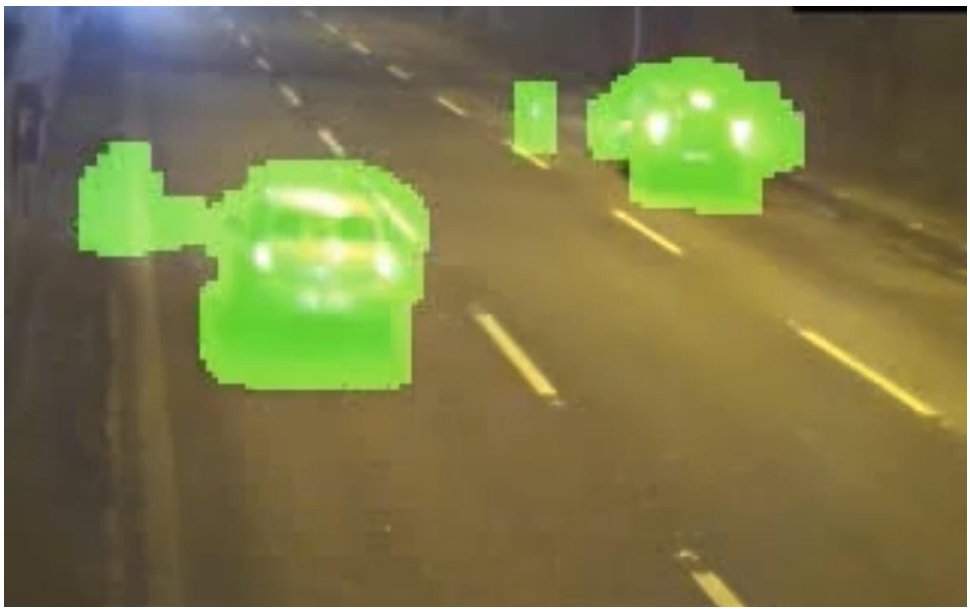


Figure 6: A state of the art background subtracting output
that fails in tunnel environment

Even the state of the art computer vision algorithms consisting of background-subtraction, object detection and object tracking, fail to solve the event detection problem alone. The noise from the low-level layers increases the error rates of the overall system, and creates unacceptable results in the final event detection task. An example of these tracking mistakes can be seen in Figures 1 and 2, where the former is a background subtracting mistake and the latter is a tracking error. It can be seen that the foreground mask marks areas out of the cars in Figure 1 and the tracking

trajectory of the vehicle has deviated dramatically from the original route, which is actually linear.

Figure 7: A state of the art tracking output that fails in the tunnel environment

3.1 Theory

Statistical pattern recognition methods such as Hidden Markov Models (HMM) have been widely studied for short-term events such as human gesture. However, if the event is spread over a long time and involves complex activities it is difficult to apply the same approach. One of the main reasons for this difficulty is that there is very limited training data compared with the massive dimensionality that is needed to accomplish successful training. The second reason for this difficulty is that most of the activities may be semantically equivalent but differ in vast amounts in feature values. These activities are much easier to detect if the activity involved preliminary domain knowledge about the nature of the events.

3.2 Symbolic Representation and Attribute Grammars

The syntactic parser in this section acts as a categorization engine as in the cognitive science literature. It reads the input and decides if the given syntax fits a predefined grammar and parses it. The parser indicates, in which category the given syntax and the occurring event fall.

Syntactic Pattern Recognition scans a series of symbols and detects specific formations occurring in the stream. These symbols are selected from a finite number of symbols, which can be terminal or nonterminal. Terminal symbols are defined as primitive events that are generated from the video. These primitive events can be a car appearing in the scene or a car changing lanes. Non-terminal symbols are non-primitive events that can inherit multiple primitive events or higher abstractions.

This approach can be limiting if additional information about the objects such as velocity or position are needed to understand the event. Attribute grammars are used when objects should carry additional informational attributes within the symbolic representation. These kinds of grammars are used in compilers and computer aided language representations. Any information that is needed to pass from lower layers to higher layers and could be detected by low level detectors can be used as attributes.

In the video domain occurrences of particular events in a particular location can have a dramatic difference compared to other locations. Having the opportunity to add unbounded number of attributes to a symbol reduces the bottlenecks created by the symbolic mechanism, which naturally sits on the top of a limited set of symbols. These symbols are parallel to concepts in the cognitive science literature. They are similar to concepts because they create an information storage unit that can be created or consumed in both abstract levels of processing in the model, namely the image processing and the parsing layers.

Attribute Grammars have been used in syntactic pattern recognition and video domains since; first introduced by Knuth (1990). Attribute grammars (AG) consist of a 5 tuple of;

$$AG= (G, SD, AD, R, C)$$

where $G = (V_N, V_T, P, S)$ is the underlying context free grammar. V_N and V_T represent the non-terminal and terminal symbols, respectively. P stands for the production rules and *S* represents the starting symbol. *SD* represents a semantic domain where functions and variables are defined and operated on these symbols. *AD* represents the attributes that can be attached to symbols. These symbols occur in the productions and each attribute is of a single type. *R* is a set of attribute evaluation rules for each $p \in P$. These rules define how the functions are defined in the semantic domain manipulate or evaluate the attributes in the production rules. *C* is the set of semantic conditions associated with $p \in P$.

Semantic conditions impose limits on the values of attributes when production rules are being used. The predicates in the semantic domain are converted to real values by the functions in *R*. These real values are constrained to a set of soft or hard limits. The outputs of these functions are bounded limits, which are either soft or hard. Soft limits are continuous and extended to values between zero and one. Hard limits mean that their value is levelled by a threshold value. If the threshold is reached the output is one, if not zero. This approach leads us to create a series of productions to represent the conjunction of constraints.

In this thesis, the following notation is used for defining attribute grammars. Words with capital letters represent nonterminal states, whereas words with lowercase letters represent terminal states. An attribute which is associated with a symbol with index number i is represented as X_i. For example, if an attribute called location belonging to an object car with index of y is a function of the location of another car

with index t, the representation will be represented as $Car1_y$.location = f ($Car2_y$.location).

In the grammatical rules X0 represents the left-hand side of the operator. X1 represents the right-hand side of the rule. If the right-hand side has multiple symbols, X1 will represent the first one, X2 will represent the second one and so on.

3.3 Recognizing Events by Parsing

Parsing the symbolic information into events is another challenge for an event detection system. For this system, the events should be detected in real time to prevent accidents; hence the parser should have a low complexity to reduce the detection delays. The parsing algorithm should effectively handle attributes that are attached to symbols. Since the system should work in real time, the parser cannot have information about symbols that might occur later in time.

3.3.1 Earley's Parser

Earley's algorithm is a top-down dynamic programming algorithm that can handle the requirements given above. It runs at n^3 complexity where n is the number of symbols (Earley, 1970). Earley's parser reads symbols from left to right sequentially and creates a list of all pending possible derivations that comply with the current terminal symbol. This list serves as a potential search table for possible parses.

In Earley's dot notation, given a production rule of $X \rightarrow \alpha\beta$, $X \rightarrow \alpha \bullet \beta$ represents a string that is being parsed where α has already been parsed and β is expected to be

parsed. When a string is being parsed, at every token between symbols the parser generates a state set originated from the last symbol parsed. According to the terminal or nonterminal nature of the last originating symbol the parser carries out any of the following actions:

- **Prediction:** For every state set in S(*k*) originated from the symbol index *j,* which follows the production rule format ($X \rightarrow \alpha \bullet Y\ \beta, j$) , if the next symbol after j is a non-terminal, evaluate all possible parses that have the form add ($Y \rightarrow \bullet\ \gamma, k$) to S(*k*) state set for every rule that has a grammatical expansion that has Y on the left side ($Y \rightarrow \gamma$). Also, since the grammar is an attribute grammar all the attributes inherited from α should be passed to the possible extended parses that has Y on the left side.
- **Scanning:** If the next input symbol in the string to be parsed is *a* which is a terminal symbol, for every state in S(*k*) of the form ($X \rightarrow \alpha \bullet a\ \beta, j$), ($X \rightarrow \alpha\ a \bullet \beta, j$) is added to S(*k*+1). Then the index of the state set *k* is incremented to *k+1* and the upcoming symbol is added to the parsed list by placing it to the left of the parsing dot • . Because the grammar is an attribute grammar, all the attributes arising from the symbol *a* should be evaluated and transferred to the next parse X in state set S(k+1).
- **Completion:** If the parsing dot is in the rightmost position and if all the evaluated attributes in the followed grammar satisfy the conditions that are previously defined, evaluate all synthesized attributes. For every state in S(k) of the form ($X \rightarrow \gamma \bullet, j$), find every state of the form ($Y \rightarrow \alpha \bullet X\ \beta, i$) and add ($Y \rightarrow \alpha\ X \bullet \beta, i$) to S(*k*). Assign all new synthesized attributes to Y's on the left hand side.

This list of actions is repeated until no more states can be added to the set. Upon completion of a parse, the algorithm outputs the parsed event. In Earley's original parser there was a symbol called look-ahead in the state set but since it had little practical effect in parsing it was dropped later. The classical Earley parser works on

a single stream of symbols. To manage multiple objects or multiple events occurring at the same time multiple parsers should be used on different streams.

CHAPTER 4

IMPLEMENTATION AND EXPERIMENTS

In this thesis an attribute context free grammar will be used to detect critical events in a tunnel surveillance scenario. The system should detect a stopped vehicle and a vehicle travelling in the wrong direction. An attribute grammar is used in order to involve the position information to verify event detection targets.

The environment in a tunnel scenario is not so complex in terms of meaning. In this study most of the events that should trigger an alarm in the warning system are handled. Objects that are detected and create primitive events are people and vehicles in the tunnel. Vehicles should enter the tunnel in the lower end of the field of view (FOV) and exit in the higher end of the FOV. The vehicles should not stop or move backwards in a safe tunnel environment. Personnel can walk through the walkway on the left for maintenance reasons but they cannot walk on the road. Similarly, if the tunnel is open for traffic and someone is on the walkway that is a dangerous situation and an alarm should be raised. Changing lanes is also a critical warning situation in most tunnels because overtaking and exceeding the speed limit is a violation of the rules. As a precaution vehicles changing lanes are also detected as events.

The system is designed to work as two layers. The first layer being the image-processing layer, which reads the video stream frame by frame and creates meaningful atomic symbols. The second layer being the grammatical parser, which reads the input atomic symbols and parses the input string accordingly and detects events.

4.1 General System Architecture

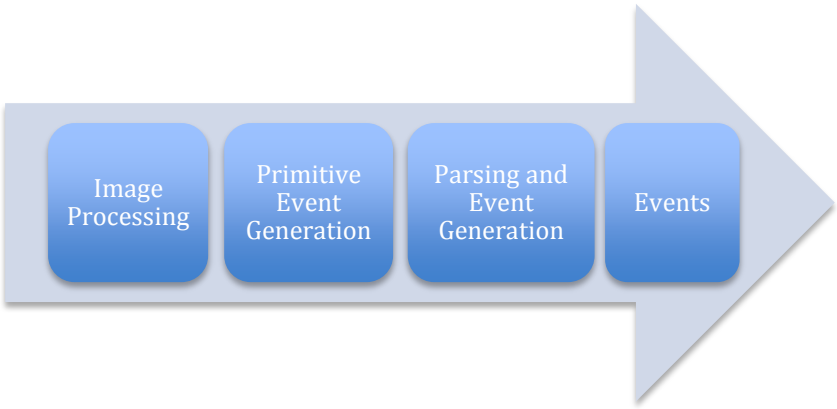


Figure 8: General system architecture

The tunnel videos to be used consist of three lanes in which vehicles are allowed to travel in one direction only. The tunnel video datasets consist of events of vehicles passing through the tunnel without any event (normal event), vehicles that stop in the middle of the tunnel (abnormal event), vehicles that travel in the reverse direction (abnormal event), people on the walkway (normal event), people on the walkway when there is a car passing (abnormal event), people walking on the road (abnormal event) and vehicles changing lanes when driving (abnormal event).

The videos are taken from a real tunnel with different lighting conditions and real-life cases. The resolution of video streams is 320 x 240, the color space is RGB and the frame rate per second is 15. The image processing software is coded using OpenCV library and for Earley's parser a c ready implementation is used but modified to handle attribute grammars.

The first layer of the software is the image-processing layer, which has the responsibility to create a stream of symbols and attributes for each vehicle throughout its journey. This software consists of five main blocks; background subtracting, shadow removal, connected component analysis, tracking, symbol and event generation.

The second layer of the software is the parser and event detection, which has the responsibility to read the input stream of symbols and attached attributes, parse the stream with the rules of the context free attribute grammar and create events accordingly. An Earley parser with attribute grammar modifications is used for this purpose. The outputs of the parser are the abnormal and normal events. Abnormal events are; stopped vehicle, wrong way vehicle, vehicle changing lanes, person existence while a car exists, person entering the road and the normal events as vehicle passing and person passing on aisles without breaking any rules.

4.2 Image Processing Layer

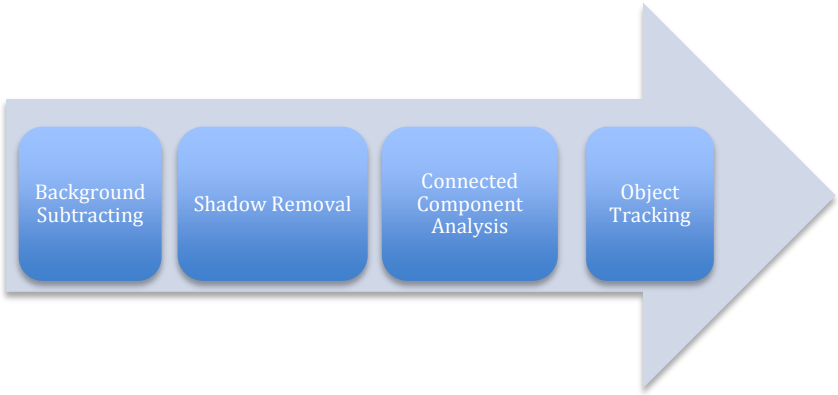


Figure 9 : Breakdown of image processing algorithms

Background detection layer mainly labels pixels that are temporally and spatially different than the memorized background model. For background subtracting purposes MoG (Stauffer and Grimson, 1999), Codebook proposed by Kim et al., (Kim and Chalidabhongse, 2005) and Lehigh Omnidirectional Tracking System (LOTS) (Boult, 1999) are evaluated and LOTS background subtracting method is used because of its superiority in accuracy and processing performance.

For shadow detection, a shadow detection algorithm based on a color based shadow reduction algorithm (Cucchiara, 2003) and a texture-based algorithm (Grest et al., 2003) are considered. By reviewing the previous comparisons of these algorithms in Monteneiro‘s work, the color normalized cross-correlation method created by Grest, incorporating texture-based and color-based approaches is chosen and implemented. For the connected component analysis a linear connected component labelling algorithm (Chang, 2004) is used. The code is taken from the OpenCV library because its superiority in terms of accuracy and performance.

Figure 10. Results of separate image processing layers a) Raw image b) Result of background subtraction (BGS) d) Result of shadow reduction d) Result of tracking

For the purposes of tracking the Optical Flow tracking algorithm (Lucas and Kanade, 1981) is used. This tracker finds feature points in the connected blob region then finds the corresponding feature points in the next frame. By grouping these feature vectors with the RANSAC algorithm (Fischler, 1981) the spatio-temporal locations of every object are detected and copied to a trajectory storage belonging to each object over time.

A simple moving averaging filter is used to rectify these tracking trajectory results to minimize deviations from ground truth trajectories. These deviations are mainly caused by the ineffectiveness of the feature points matching algorithm, and the feature vector summarization steps in RANSAC algorithm.

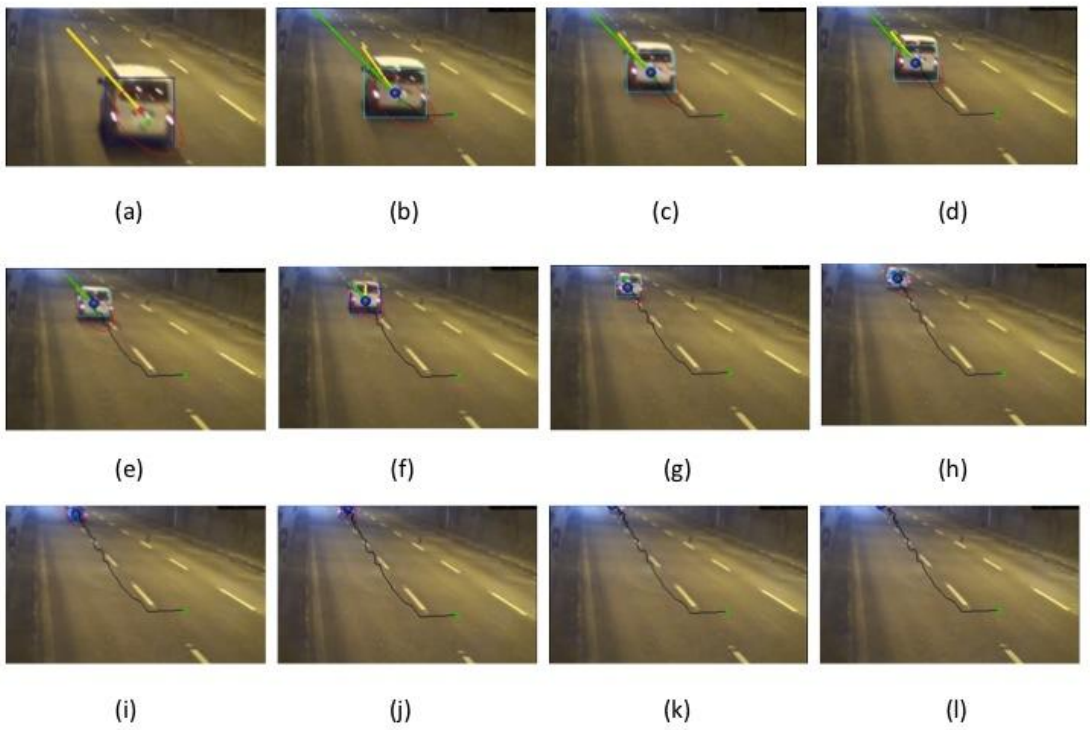


Figure 11: Successful tracking of a car during the consecutive frames a-l using the optical flow method

4.3 Primitive Event Generation

The primitive events generated by the system are: *car_appear*, *car_disappear*, *car_stopped*, *car_moving_further*, *car_moving_closer, person appear, person disappear, person moving.* These primitive events, which provide the basis for the grammar, are explained by definitions and the related methods used to produce them.

The functions creating these events are:

4.3.1 Car_appear:

Car_appear: This represents the appearance of a car in any place in the field of view. The car can appear at the far end or closer end of the tunnel as well as at any point in the tunnel resulting from tracking mistakes. This atomic event is created when the tracking layer decides that an object is a car by measuring its size and aspect ratio.

4.3.2 Person_appear:

Person_appear: This represents the appearance of a person at any point in the field of view. The person can appear at the far end or closer end of the tunnel as well as any place in the tunnel resulting from tracking mistakes. This atomic event is created when the tracking layer decides that an object is a person by measuring its size and aspect ratio.

If a blob (classified as a person or car by the visual recognizer) occurs in the tracking subsection of the image-processing layer and this blob begins to be tracked, an object entity is created, thus a primitive event for car_appear is generated with this function. If an object is created and tracked for a minimum of three frames, then the event is generated. This value is found empirically and designed to eliminate noise created from connected component layer. If this number is one or two the tracking creates many false alarms. If this number is larger than three the object is not created for this number of frames this leads to an equal latency in object creation.

4.3.3 Car_disappear:

Car_disappear: This represents the disappearance of a car at any point in the field of view. It can disappear at the far end or closer end of the tunnel as well as at any place in the tunnel resulting from tracking mistakes. This atomic event is created when tracker decides that the tracked object, which is a car, does not match any current blob in the frame.

4.3.4 Person_disappear:

Person_disappear: This represents the disappearance of a person at any point in the field of view. It can disappear at the far end or closer end of the tunnel as well as at any place of the tunnel resulting from tracking mistakes. This atomic event is created when tracker decides that the tracked object, which is a person, does not match any current blob in the frame.

4.3.5 Car_stopped:

Car_stopped: This represents a car stopping a predefined amount of time, within a predefined area. To create the primitive event car_stopped, the object trajectory is searched against the current position difference with the past positions for 15 frames and if the distance between the current location and the past locations stays under a predefined limit for all of these frames, the object is assumed to stop moving and a

car_stopped event is generated. The object attribute for moving is also labelled as not moving after this event is created.

4.3.6 Car_moving_further & Car_moving_closer:

The primitive events car_moving_further and car_moving_closer are detected in the same function to reduce computational complexity.

Car_moving_further: This represents a car travelling in the allowed direction of the tunnel. The movements in this category are from close to the start of the tunnel to the far end of the tunnel.

Car_moving_closer: This represents a car travelling in the disallowed direction of the tunnel. The movements in this category are from the far end of the tunnel to close to the start of the tunnel.

The Car_Moving_Further_Or_Closer() function takes the input of a initialized object and calls an event for car_moving_further if the car is moving forward, it calls an event for car_moving_closer if the car is moving backwards. It first reads the current position of the object and then generates a voting mechanism to decide if the car is moving backwards or forwards. The current position of the object is compared to the three previous, six previous and nine previous positions respectively and three votes of whether a backward or forward direction is obtained. Then the first previous location is compared to the four previous, seven previous and the previous positions of the object and three more votes are obtained. Then the second previous position of the object is compared to the five previous, eight previous and eleven previous positions of the object and three more votes are obtained. By this iterative voting technique, a more noise-free understanding of the motion was reached and the system becomes more sustainable to three tracking noise. The jumping distance of

three was empirically found. At the end of the function, all the votes are compared and if the function voted for forward more times then the forward direction is selected. Conversely if the function votes for backwards, the backward direction is selected as the motion direction of the vehicle.

4.3.8 Person_moving:

Person_moving: This represents a person moving in any direction of the tunnel. The movements in this category can range from far end of the tunnel to close to the start of the tunnel. If the person object that is tracked moves its position more than a predefined threshold, in a predefined number of frames, an atomic action of person moving is created.

4.4 Parsing and Event Generation

The primitive events are defined in Chapter 3. The attribute that is attached to these events is location, which leads to computation of the lane information.

Table 7: Grammar rules for the proposed parsing algorithm –

The left column refers to the grammar rules, the right column refers to the attribute rules

TUNNEL_EVENT -> CAR_TRANSPASSING | CAR_WRONG_WAY | CAR_STOPPED | PERSON_ON_THE_ROAD | PERSON_WHILE_CAR | CAR_CHANGE_LANE	
---------- CAR_TRANSPASSING -> car_appear CAR_MOVING_FURTHER car_disappear	---------- isEqual(X1.lane,X2.lane) isEqual(X1.lane,X3.lane) ----------
---------- CAR_WRONG_WAY -> car_appear CAR_MOVING_CLOSER	
CAR_WRONG_WAY -> car_appear CAR_MOVING_CLOSER car_disappear	
CAR_WRONG_WAY -> car_appear CAR_MOVING_FURTHER CAR_STOPPED CAR_MOVING_CLOSER	
CAR_WRONG_WAY -> car_appear CAR_MOVING_FURTHER CAR_STOPPED CAR_MOVING_CLOSER car_disappear	
CAR_WRONG_WAY -> car_appear CAR_MOVING_FURTHER CAR_MOVING_CLOSER	
CAR_WRONG_WAY -> car_appear CAR_MOVING_FURTHER CAR_MOVING_CLOSER car_disappear	
---------- CAR_MOVING_FURTHER -> car_moving_further	---------- XO.lane:= X1.lane
---------- CAR_MOVING_FURTHER -> CAR_MOVING_FURTHER car_moving_further	---------- isEqual(X1.lane,X2.lane) XO.lane:= X1.lane
---------- CAR_CHANGE_LANE -> car_appear CAR_MOVING_FURTHER car_moving_further	---------- ~isEqual(X1.lane,X2.lane) || ~isEqual(X2.lane,X3.lane)
---------- CAR_CHANGE_LANE -> car_appear CAR_MOVING_FURTHER	---------- ~isEqual(X2.lane,X3.lane) || ~isEqual(X3.lane,X4.lane) ||

car_moving_further car_disappear	~isEqual(X4.lane,X5.lane)
--- CAR_MOVING_CLOSER -> car_moving_closer CAR_MOVING_CLOSER -> CAR_MOVING_CLOSER car_moving_closer	---
--- CAR_STOPPED -> car_stopped CAR_STOPPED -> CAR_STOPPED car_stopped	--- X0.lane:=X1.lane X0.lane:=X2.lane
--- CAR_STOPPED -> car_appear CAR_STOPPED CAR_STOPPED -> car_appear CAR_MOVING CAR_STOPPED	--- X0.lane:=X2.lane X0.lane:=X3.lane
--- CAR_MOVING -> CAR_MOVING_FURTHER CAR_MOVING -> CAR_MOVING_CLOSER	--- X0.lane:=X1.lane X0.lane:=X1.lane
--- PERSON_TRANSPASSING -> person_appear PERSON_MOVING person_disappear	--- isEqual(X1.lane,0)& isEqual(X2.lane,0)& isEqual(X3.lane,0)
--- PERSON_ON_THE_ROAD -> person_appear PERSON_MOVING person_disappear	--- !isEqual(X1.lane,0)|| !isEqual(X2.lane,0)|| !isEqual(X3.lane,0)
--- PERSON_MOVING -> person_moving PERSON_MOVING -> PERSON_MOVING person_moving	--- X0.lane:=X1.lane X0.lane:=X2.lane
PERSON_WHILE_CAR -> person_appear PERSON_MOVING car_appear PERSON_WHILE_CAR -> car_appear CAR_MOVING person_appear PERSON_WHILE_CAR -> person_appear PERSON_MOVING car_appear CAR_MOVING	---

PERSON_WHILE_CAR -> car_appear CAR_MOVING person_appear PERSON_MOVING PERSON_WHILE_CAR -> car_appear person_appear CAR_MOVING PERSON_MOVING PERSON_WHILE_CAR -> car_appear person_appear PERSON_MOVING CAR_MOVING	

The event CAR_TRANSPASSING means that a car is passing through the tunnel without any significant abnormality such as stopping or moving backwards. It is defined as a constant motion from the start of the tunnel to the end. There is no specific attribute condition on CAR_TRANSPASSING.

CAR_WRONG_WAY is defined as any motion by a car in reverse to the traffic direction of the tunnel. This event can be caused by a car moving directly in the reverse direction after appearing or a car moving first in the right direction and then stopping and then starting to move in the wrong direction.

CAR_STOPPED is any action caused by a car ending with stopping. It can be an event that is directly caused by stopping after the car is observed. It may also happen if a car has first moved in the forward or backward direction and stopped afterwards.

CAR_MOVING is any action of the three which are CAR_MOVING_FURTHER , CAR_MOVING_CLOSER or CAR_CHANGING_LANE. The first two actions are actions that are performed in the same lane whereas the third action is an action in which car moves in the further direction but changes lane. To detect these events the attributes of the lane are constantly monitored by the parser.

There are four non-terminals referring to actions of people, which are PERSON_TRANSPASSING, PERSON_MOVING, PERSON_WHILE_CAR and

PERSON_ON_THE_ROAD. PERSON_TRANSPASSING is a normal event where a person is walking on the walkway when there is no car in the scene.

PERSON_ON_THE_ROAD is a dangerous and abnormal event where person enters the traffic zone from the walkway. PERSON_WHILE_CAR is an event where person is on the walkway but a car is present in the FOV, thus it is a critical and abnormal situation where danger exists. PERSON_MOVING refers to a state where the person is moving whether on the walkway or from the walkway to the road. To detect these situations the parser continuously monitors attributes of lane changes.

CHAPTER 5

RESULTS AND DISCUSSION

5.1 Dataset

The experiments are performed on 12 video datasets consisting of a total of 21:41 seconds and 19,515 frames. The videos contain all the normal and abnormal events mentioned.

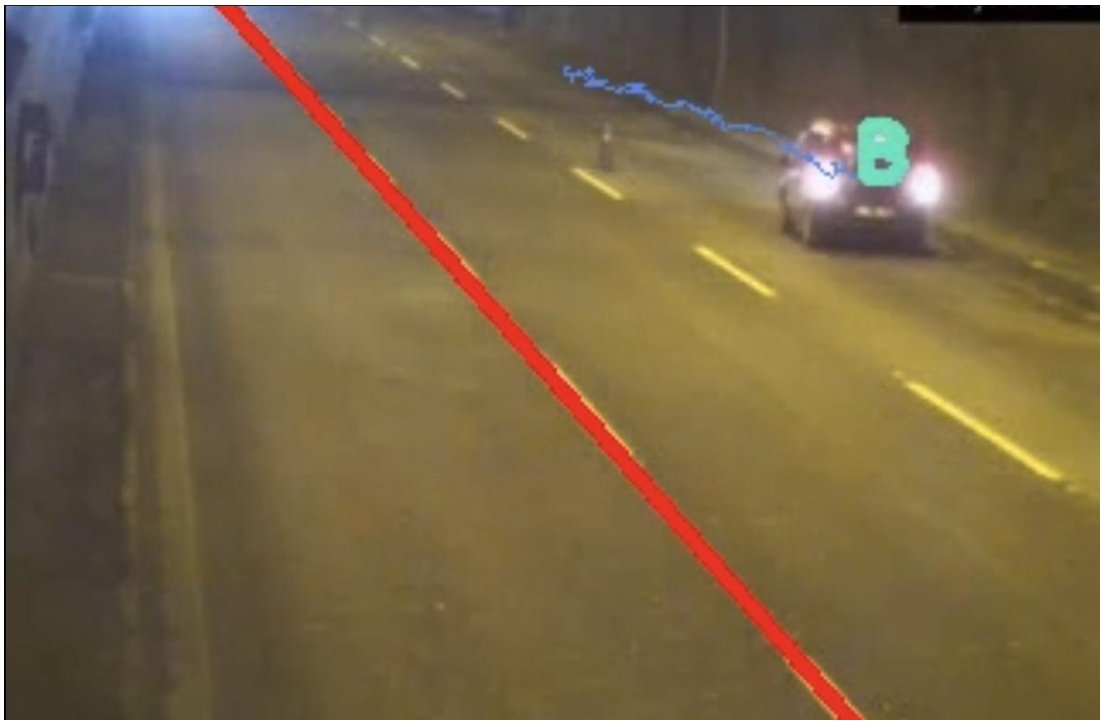


Figure 12: An event that is labelled as a car moving in the wrong direction

Of the 12 videos there are 59 CAR_TRANSPASSING events. This is considered to be a normal event and even if it is parsed it will not raise an alarm condition. There are seven WRONG_WAY, 26 CAR_STOPPED, 14 PERSON_WHILE_CAR, 44 PERSON_ON_ROAD, 59 CAR_TRANSPASSED, and four CAR_CHANGE_LANE events.

5.2 Results of the Attribute Grammar

In Table 8 the parser detection performance of 'Wrong Way' events is shown. The total number of occurrences, total number of correct classifications, total number of missing and total number of misdetections of 'Wrong Way' events are presented. The results show that four of seven events are correctly classified and no misdetections have occurred.

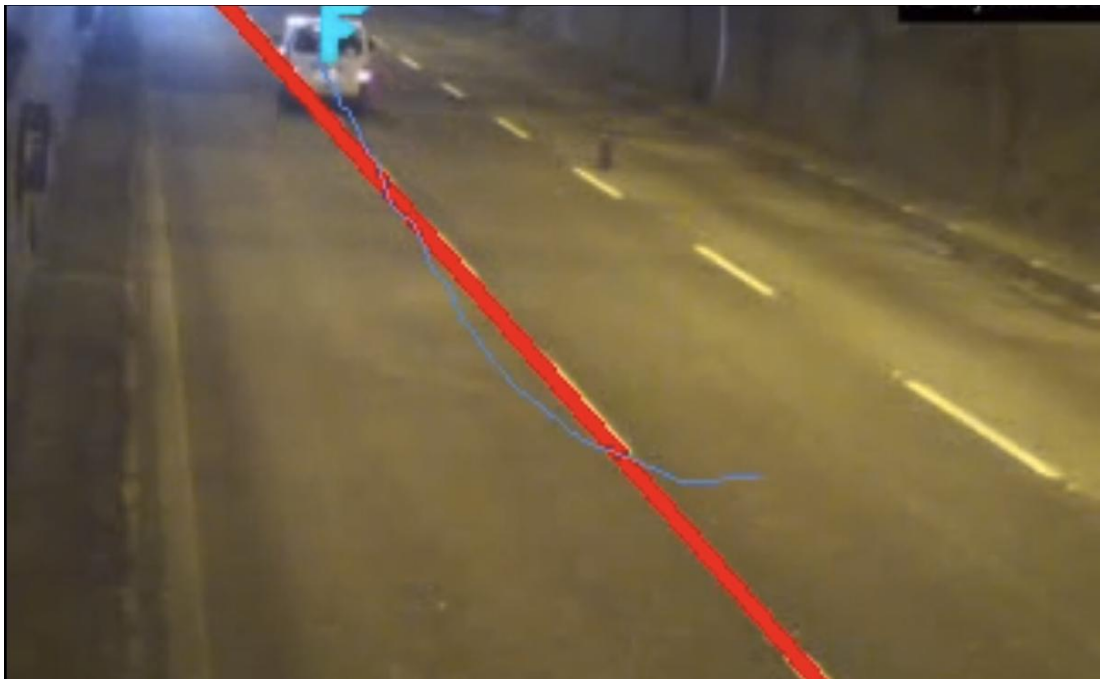


Figure 13: An event labelled as a car changing lanes

Table 8: Detection results for wrong way events

	Total # of Wrong Way Events	Total # of Correctly Detected Wrong Way Events	Total # of Missed Wrong Way Events	Total # of Misdetected Wrong Way Events
Videos_all	7	4	3	0

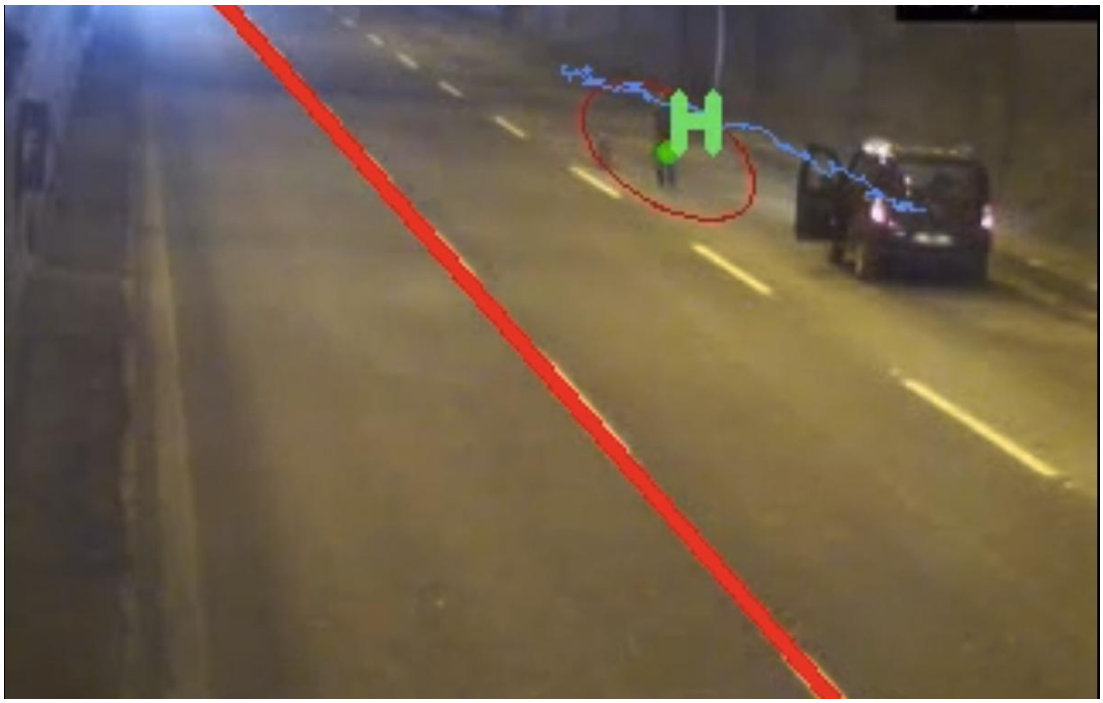


Figure 14: An event labelled as a person on the road

Table 9: Detection results for car stopped events

	Total # of Car Stopped Events	Total # of Correctly Detected Car Stopped Events	Total # of Missed Car Stopped Events	Total # of Misdetected Car Stopped Events
Videos_all	26	22	4	0

The parser detection performance of ‘Car Stopped’ events is shown in Table 9. The total number of occurrences, total number of correct classifications, total number of missing and total number of misdetections of ‘Car Stopped’ events are presented. The results show that 22 out of 26 events are correctly classified and no misdetections have occurred.

Table 10: Detection results for person while car events

	Total # of Person While Car	Total # of Correctly Detected Person While Car Events	Total # of Missed Person While Car Events	Total # of Misdetected Person While Car Events
Videos_all	14	11	3	8

Table 10 gives the parser detection performance of ‘Person While Car’ events. The total number of occurrences, total number of correct classifications, total number of missing and total number of misdetections of ‘Person While Car’ events are presented. The results show that 11 out of 14 events are correctly classified and 8 misdetections have occurred.

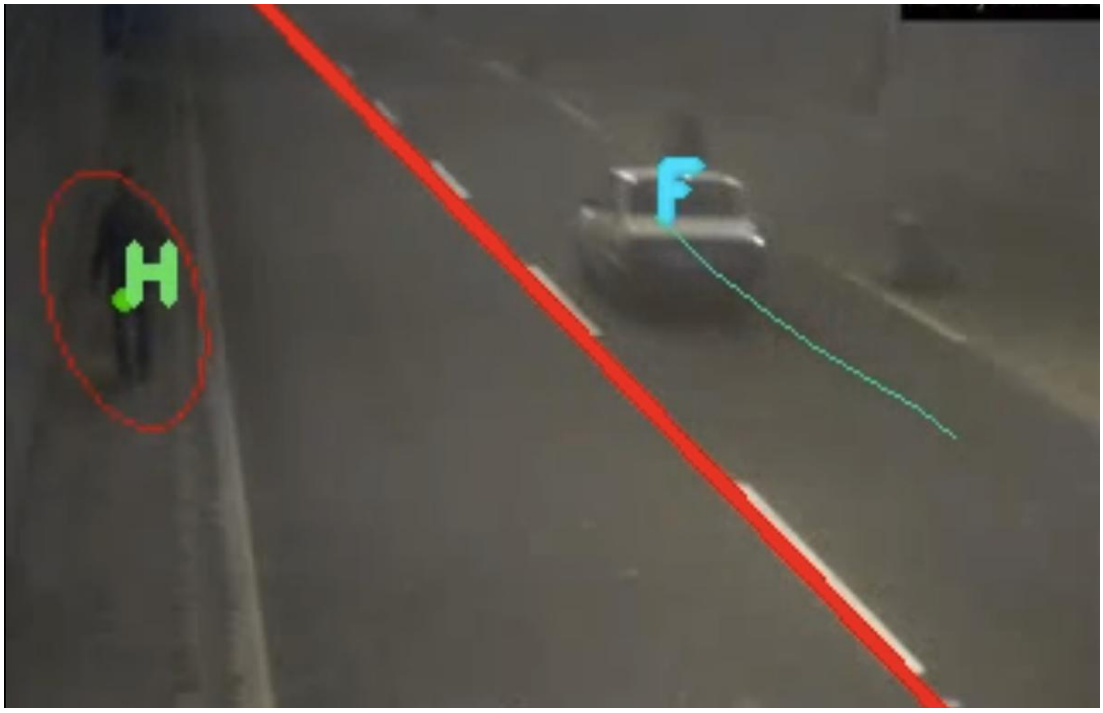


Figure 15: An event labelled as a person while car

Table 11: Detection results for person on road events

	Total # of Person on Road Events	Total # of Correctly Detected Person on Road Events	Total # of Missed Person on Road Events	Total # of Misdetected Person on Road Events
Videos_all	44	43	1	4

In Table 11 are shown the parser detection performance of 'Person on Road' events. The total number of occurrences, total number of correct classifications, total number of missing and total number of misdetections of 'Person on Road' events are presented. The results show that 43 out of 44 person on road events are correctly classified and 4 misdetections have occurred.

Table 12: Detection results for car change lane events

	Total # of Car Change Lane Events	Total # of Correctly Detected Car Change Lane Events	Total # of Missed Car Change Lane Events	Total # of Misdetected Car Change Lane Events
Videos_all	4	4	0	0

The parser detection performance of 'Car Lane Change' events is shown in Table 12. The total number of occurrences, total number of correct classifications, total number of missing and total number of misdetections of 'Car Lane Change' events are presented. The results show all 4 events are correctly classified and no misdetections have occurred.

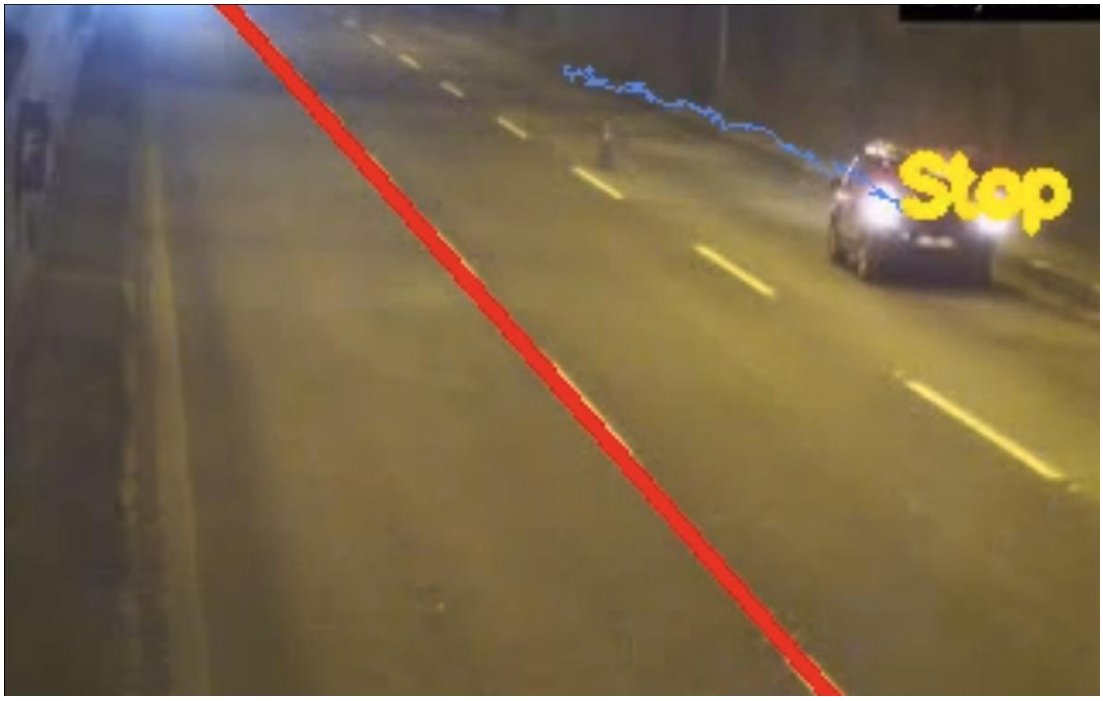


Figure 16: An event labelled as a car stopped

Table 13: Detection results for car transpassed

	Total # of Car Transpassed	Total # of Correctly Detected Car Transpassed	Total # of Missed Car Transpassed	Total # of Misdetected Car Transpassed
Videos_all	59	57	2	6

In Table 13 is the parser detection performance of 'Car Transpassed' events. This event represents a safe passage by a car through the tunnel with no rules being broken so no alarm should be raised. The total number of occurrences, total number of correct classifications, total number of missing and total number of misdetections of 'Car Transpassed' events are presented. The results show 57 out of 59 events are correctly classified and 6 misdetections have occurred.

Table14: Summary of detected, missed and misdetected events

	Total # of Events	Total # of Detected Events	Total # of Missed Events	Total # of Misdetected Events
Videos_all	154	141	13	18
Percentage to Total # of Events	100%	91.55%	8.45%	11.68%

Table 14 presents the general parser detection performance of all events. The total number of occurrences, total number of correct classifications, total number of missing and total number of misdetections of events are presented. The parser could detect 141 of the 154 events whereas 18 misdetections have occurred. As a

percentage, 91.55% out of the total events are detected whereas 11.68% of events are misdetected.

The results show that the attribute grammar that was followed can detect most of the events in the dataset. One reason for this success is that the datasets do not contain ambiguous situations as headlights, and trucks blocking the scene. Another reason is that the deterministic grammar was chosen and updated to fit the data model recursively during the design. However, the parser faces some challenges in terms of misdetection which are due to person events and car transpassed events. When a person enters the scene whether on the walkway or in the road area, the tracker performs poorly on the target, losing the person and relocates it a number of times. Even assuming that a real person triggers the person event, actually that event has occurred once but the system raises multiple alerts. These results clearly show attribute grammars can be used to parse event sequences in tunnel videos.

To further clarify the advantages of using an attribute grammar as an event detector, a second set of results is enlightening. Table 15 shows the outputs of the event detector system and the raw outputs of the primitive event generator system.

Table 15: Summary table of results with primitive events included

	Total # of Events	Total # of Outputs of Event Detector	Total # of Missed Events	Total # of Misdetected Events	Total # of Related Primitive Events
Wrong Way	7	4	3	0	458
Car Stopped	26	22	4	0	853
Person While Car	14	11	3	8	231
Person On Road	44	43	1	4	1231
Change Lane	4	4	0	0	53
Car Transpassed	59	57	2	6	1544
Total #	154	141	13	18	4370

According to Table 15, the grammar detected all events accurately. All the separate events are correctly classified. However, another important output of the algorithm is it compresses a great deal of information. For example, a wrong way event has been created 458 times during all the video sequences. If the grammar had not been used and these outputs would have created alerts directly thus the system would have raise 458 alerts, when only seven cars violated the wrong way law. Similarly, a car stopped primitive event is created 853 times whereas there is only 26 cars stopping in the sequences. Person while car, person on road, change lane and car transpassed events are created 231, 1, 231, 53, and 1,544 times respectively during the sequences whereas in reality only 14 person while car, 44 person on the road, four change lane and 59 car transpassed events occurred.

Table 16: Compression ratio table

	Total # of Events	Total # of Detected Events	Total # of Missed Events	Information Loss Ratio	Total # of Related Primitive Events	Compression Ratio
Wrong Way	7	4	3	42.85 %	458	99.12%
Car Stopped	26	22	4	15.38 %	853	97.42%
Person While Car	14	11	3	21.42 %	231	95.23%
Person On Road	44	43	1	2.27 %	1231	96.50%
Change Lane	4	4	0	0 %	53	92.45%
Car Transpassed	59	57	2	3.38 %	1544	96.30%
Overall	154	141	13	8.44%	4370	96.77%

Table 16 displays the results of the grammar from a different approach presenting them as a compression mechanism for the raw data. The model is inspired by a vision system referred to by various cognitive scientists (Marr 1982 and Fodor 1975). The presented output can be an implication of the mental reasoning for a complex event that is triggered by a visual input. If the low-level detectors are abstracted as they can undertake the data summarization from raw images to primitive events whereas attribute grammars can model the data summarization from primitive events to real events. The compression ratio is calculated by dividing the number of detected events by the number of related primitive events and subtracting this number from one. The information loss ratio is calculated by dividing the number of missed events by the number of real events. The system achieves an overall compression rate of 96.77% for 4,370 primitive events. The system loses 8.44% of the information, which comprise lost events due to errors.

The dataset includes a variety of events with a number as high as 154 for real events and 4,370 for primitive events. Considering that the number of stopped vehicle or wrong way events is very rare in tunnels, this dataset can be considered adequate.

The dataset was acquired from 8 cameras in 2 of the Adana Bahçe Tunnels. Since there is a diversity in angles and positions of the cameras, the dataset can be considered to be sparse and the results can be considered to be obtainable similarly from different datasets from cameras in other tunnels.

5.3 Results of SVM Implementation

To objectively evaluate the output of the attribute grammar, a SVM implementation of the detector is implemented. This creates a multi-dimensional map from the input data, which are the primitive events. The SVM implementation results in outputs that match the same patterns as obtained by the attribute grammar. So they are comparable with each other.

The SVM implementation takes the input of primitive event arrays and searches for patterns of events in window lengths of 32 units. These 32 unit arrays are fed to the SVM and SVM outputs a final decision as to whether the event is a previously defined event or there is no event. The implementation uses libSVM and there is a separate training set obtained from a different dataset taken from the same tunnel. The person while car event cannot be tested because the implementation of SVM cannot handle interactions by multiple objects. Thus, the results of the two systems are compared with the person while car event excluded.

Table 17: Summary table of SVM results with primitive events included

	Total # of Events	Total # of Outputs of Event Detector	Total # of Missed Events	Total # of Misdetected Events	Total # of Related Primitive Events
Wrong Way	7	2	5	0	458
Car Stopped	26	16	10	0	853
Person While Car	--	--	--	--	---
Person On Road	44	33	9	4	1231
Change Lane	4	2	2	8	53
Car Transpassed	59	30	29	12	1544
Total #	140	83	55	24	3915

The outputs for SVM for events other than 'Person While Car' is given in Table 17. That event person while car is excluded because of the inadequacies of SVM. The results show that 83 of 140 events are correctly classified and 55 are missed and 24 are misdetected throughout the sequence.

Table 18: Summary of the mixed results with primitive events included

	Total # of Events	Total # of True Detections (AG/SVM)	Total # of Missed Events (AG/SVM)	Total # of Misdetected Events (AG/SVM)	Total # of Related Primitive Events
Wrong Way	7	4/2	3/5	0/0	458
Car Stopped	26	22/16	4/10	0/0	853
Person While Car	--	--	--	--	---
Person On Road	44	43/33	1/9	4/4	1231
Change Lane	4	4/2	0/2	0/8	53
Car Transpassed	59	57/30	2/29	6/12	1544
Total #	140	130/83	10/55	10/24	3915

Table 18 shows the results of SVM versus attribute grammar (AG) performance on the dataset. The AG performs better detecting 130 events as opposed to 83 for the SVM. It also performs better on the missed events (10 versus 55) and much better on misdetections (10 versus 24).

Table 19: Summary of SVM performance versus attribute grammar (AG) results with correct detection ratio, information loss and compression ratio

	Total# of Primitive Events,	Total # of Events	Total # of Correctly Detected Events	Total # of Missed Events	Correct Detection Ratio	Information Loss Ratio	Compression Ratio
AG	3915	140	130	10	92.85%	7.14%	96.68%
SVM	3915	140	83	55	59.28%	39.28%	97.88%

As clearly shown in Table 19 the AG performs much better in detection performance thus information loss criteria as 130 out of 140 events are detected by the AG and only 83 events are detected in SVM. The compression ratios 96.68% and 97.88% for the AG and SVM respectively; are very close to each other. The correct detection ratio of AG and SVM are 92.85% and 59.28% respectively. In short, the detection performance of algorithms vary widely whereas the attribute grammar results in a much better performance.

CHAPTER 6

CONCLUSION

In this thesis a system for detecting events and detecting interactions between people and cars in a tunnel environment is proposed. The approach that is used is inspired by a century of incremental study on how the human mind works with particular to the visual system. A visual event recognition model is suggested in the light of concepts, categorization, abstraction, dimensional reduction, ecological affordances, information compression and grammars.

The processing model is divided into two main blocks, one of which processes low level information and the other processes high level information. This model makes use of the abstraction principle that is a foundation of how the mind works. The information transferred between these layers consists of atomic primitive events in the context of this thesis. These events are parallel to concepts in the cognitive science literature, which is referred in the literature review section.

After these atomic primitive events are formed, the complex visual information that requires a vast amount of information is turned into a smaller and compact symbolic representation. These representations, which we refer as concepts, are fed into a grammar. Through this approach the information is compressed and a great amount of dimensional reduction is achieved. The results in the thesis prove the information is compressed to symbolic representations and a great amount of compression has been achieved, although the actual throughput of the system, in terms of event detection performance, is not changed.

The main reason behind this success is the superior performance of grammars in detecting patterns in a string of symbols. The tunnel environment can be called a *niche* in ecology terms. This environment only provides certain car and people activities. So what an observer may perceive is also restricted in this environment. These affordances as to how a car or a pedestrian can move formed the basis for the grammar and event detection. Thus, only a limited number of atomic symbols and actual events are afforded. Since the number of actual event outputs (parses) and the number of primitive events (symbols) are restricted by the affordances imposed by the *niche*, a grammatical approach is a very appropriate solution, as the experiment results indicate.

Another advantage of the system is that it is computationally inexpensive compared to other higher-level detectors such as the HMM's (Stolcke 1995, Levee 2009). Since the computational complexity is low and grammars have superior performance in this event detection problem, we may claim that if we assume that the input is a visual stream and that the environment is limited and fixed in terms of what it affords, the actual event categorization mechanism in the mind may be grammars.

The system raises alarms for events that are modeled as concepts throughout the thesis. These event categories such as a car stopping are not created from any visual physical featural entity such as color or dimension. They are created because that category was necessary in order to understand the underlying dangerous situation or the meaning behind what is happening at the tunnel in that instance. These combined categories summarize the problem in the tunnel in any situation. Since these concepts are created for functional needs, the concepts as theories definition it the best fit to the concept definition of this thesis.

The detection performance of the system is tested against SVM, which is a standard method in pattern recognition. The results show the performance of the attribute

grammar is much better in detection. Almost all of the errors emanate from the image processing layer. For most of the errors the original source is vehicles occluding each other. When a vehicle occludes any other vehicle, as long as the camera perspective is from the side of the tunnel, occlusion cannot be completely avoided. To overcome occlusion errors a better visual tracking algorithm could be used. In this thesis the visual tracking and attribute grammar are completely different layers. In future work, allowing these two layers to interact more and feedback being given from the grammar layer to the tracking layer can minimise the error rate of the total system.

As the results of these experiments are novel, the approach should be tested by other datasets to further prove the achievements of the model. Extending the attribute grammar to include other abnormalities or extending them to other scenarios is planned as future work.

REFERENCES

Boult, T. E., Micheals , R., Gao, X., Lewis, P., Power, C., Yin, W., & Erkan, A. (1999). Frame-Rate Omnidirectional Surveillance and Tracking of Camouflaged and Occluded Targets. *In Proceedings of IEEE Workshop on Visual Surveillance, USA, 2,* 48-55

Brand, M. (1996). Understanding Manipulation in Video. *Proceedings of the International Conference on Automatic Face and Gesture Recognition, USA, 2,* 94-99.

Brooks, L. R. (1978). Nonanalytic Concept Formation and Memory for Instances. *In Cognition and concepts, USA,* 169-211

Chang, F., Chen, C. J., & Lu, C. J. (2004). A Linear-Time Component-Labeling Algorithm using Contour Tracing Technique. *Computer Vision and Image Understanding, 93(2),* 206–220.

Cucchiara, R., Grana, C., Piccardi, M., & Prati, A. (2003). Detecting Moving Objects, Ghosts and Shadows in Video Streams. *In IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(10),* 1337–1342.

Cucchiara, R., Grana, C., Piccardi, M., Prati, A., & Sirotti, S. (2001). Improving Shadow Suppression in Moving Object Detection with Hsv Color Information. *Proceedings of IEEE Intelligent Transportation Systems, USA,* 334– 339.

Earley, J. (1970). An Efficient Context-Free Parsing Algorithm. *Com□munications of the ACM, 13(2),* 94–102.

Edelman, S. (2008). *Computing the Mind: How the Mind Really Works*. New York: Oxford University Press.

Fischler, M.A., & Bolles, R. C. (1981). Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Comm. of the ACM, 24 (6),* 381–395

Fodor, J. A. (1975). *The Language of Thought*. Cambridge, MA: Harvard University Press.

Fodor, J. A. (1981). *Representations: Philosophical Essays on the Foundations of Cognitive Science.* Cambridge, Mass.: The MIT Press.

Fu, K. S. (1982). *Syntactic Pattern Recognition and Applications.* New Jersey: Prentice-Hall Inc.

Gelman, S. A., & Medin, D. L. (1993). What's So Essential About Essentialism? Different Perspective on the Interaction of Perception, Language, and Conceptual Knowledge. *Cognitive Development, 8, 157-167*

Gibson J. J. (1966). The Problem of Temporal Order in Stimulation and Perception. *Journal of Psychology, 62, 141-149*

Gibson, J. J. (1966). The Senses Considered as Perceptual Systems. *Boston: Houghton Mifflin.*

Gibson, J. J. (1968). The Change From Visible to Invisible: A Study of Optical Transitions (Motion Picture Film). *Psychological Cinema Register, State College, Pa.*

Gibson, J. J. (1968). What Gives Rise to the Perception of Motion? *Psychological Review, 75, 335-346.*

Gibson, J. J. (1986). *The Ecological Approach To Visual Perception*. Hove, UK: Psychology

Press Gopnik, A., & Wellman, H.M. (1994). The Theory Theory. *In Mapping the Mind: Domain Specificity in Cognition and Culture, UK, 257-293*

Gopnik, A., & Meltzoff, A.N. (1997). *Words, Thoughts, and Theories.* Cambridge, MA: MIT Press.

Grest, D., Frahm , J. M., & Koch, R. (2003). A Color Similarity Measure for Robust Shadow

Removal in Real-Time. *In Proceedings of Vision Modeling and Visualization*, *Germany*, 253-260

Hampton, J.A., & Dubois, D. (1993). Psychology Models of Concepts: Introduction. *In Categories and Concepts: Theoretical Views and Inductive Data Analysis, ed. I. Van Mechelen, J. Hampton, R.S. Michalski, and P. Theuns, UK,* 11-33

Hull, C. L. (1920). Quantitative Aspects of the Evolution of Concepts. *Psychological Monographs, 28, 1-86*

Ivanov, Y.A., & Bobick, A. F.(2000). Recognition of Visual Activities and Interactions by Stochastic Parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 22(8), 852-872.

Joo, S. W., & Chellappa, R. (2006). Recognition of Multi-Object Events using Attribute Grammars. *International Conference on Image Processing,* 2897–2900.

Joo, S. W., & Chellappa, R. (2006). Recognition of Multi-Object Events Using Attribute Grammars. *IEEE International Conference on Image Processing,* 2897 -2900

Jurafsky, D., & Martin, J. H. (2009). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. New Jersey: Prentice-Hall Inc.

Kim, K., Chalidabhongse, T. H., Harwood, D., & Davis, L. (2005). Real-Time Foreground-Background Segmentation using Codebook Model. *In Real-time Imaging*, 11(3), 167–256.

Knuth, D. E. (1990). Attribute Grammars and Their Applications. *Lecture Notes in Computer Science, 461*, 1-12

Koffka, K. (1935). *Principles of Gestalt Psychology.* New York: Harcourt, Brace.

Komatsu, L. K. (1992). Recent Views of Conceptual Structure. *Psychological Bulletin, 112, 500-526*

Lavee, G., Rivlin, E., & Rudzsky, M. (2009). Understanding Video Events: A Survey of Methods for Automatic Interpretation of Semantic Occurences in Video. *IEEE Transactions on Systems, Man, and Cybernetics-Part C:Applications and Reviews, 39(5), 489-504*

Lucas, B., & Kanade, T. (1981). An Iterative Image Registration Technique with an Application to Stereo Vision. *In Proceedings of International Joint Conference on Artificial Intelligence*, 674–679.

Machery, E. (2009). *Doing Without Concepts*. New York: Oxford University Press.

Marr, D. & Poggio, T. (1977). From Understanding Computation to Understanding Neural Circuitry. *Neurosciences Res. Prog. Bull., 15, 470–488.*

Marr, D. (2010). *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information.* Camridge, MA: MIT Press.

Medin, D. L., & Ortony, A. (1989). Psychological Essentialism. *In Similarity and Analogical Reasoning, ed. S. Vosniadou and A. Ortony, UK, 179-195*

Medin, D. L., & Schafer, M. M. (1978). Context Theory of Classification Learning. *Psychological Review, 85, 207-238*

Mervis, C. B., & Rosch, E. (1981). Categorization of Natural Objets. *Annual Review of Psychology, 32, 89-115*

Moore, D., & Essa, I. (2002). Recognizing Multitasked Activities from Video using Stochastic Context-Free Grammar. *In Proceedings of National Conference on Artificial Intelligence,USA, 8, 770-776*

Monteiro, G. L. M. V. (2008). Traffic Video Surveillance for Automatic Incident Detection on Highways. (Master of Science dissertion , Tecnology University of Coimbra , 2008) Retrieved from http://its.isr.uc.pt/publications/MScThesis-GMonteiro.pdf

Murphy, G. L. (2002). *The Big Book of Concepts.* Cambridge, MA: MIT Press.

Newburn, T., & Hayman. S. (2001). *Policing, Surveillance and Social Control: CCTV and police monitoring of suspects*. Cullompton, Devon: Willian Publishing

Nosofsky, R. M. (1986). Attention, Similarity, and The Identification-Categorization Relationship. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 115, 39-57*

Nosofsky, R.M., & Johansen, M. K. (2000). Exemplar-Based Accounts of Multiple-System Phenomena in Perceptual Categorization. *Psychonomic Bulletin Review, 7, 375-402*

Nosofsky, R. M., & Zaki, S. R. (1998). Dissociations between Categorization and Recognition in Amnesic and Normal Individuals: An Exemplar-Based Interpretation. *Psycological Science, 9, 247-255*

Ogale, A.S., Karapurkar, A., & Aloimonos, Y.(2007) View-invariant Modeling and Recognition of Human Actions using Grammars. *ECCV Workshop on Dynamical Vision, Germany, 4358*, 115–126.

Regder, B. (2003). A Causal-Model Theory of Conceptual Representation and Categorization. *Journal of Experimental Psychology: Learning, Memory, and Cognition , 29,1141-1159*

Rips, L. J. (1995). The Current Status of the Research on Concept Combination. *Mind & Language, 10, 72-104*

Ryoo, M. S., & Aggarwal, J. K. (2006). Recognition of Composite Human Activities Through Context-Free Grammar Based Representation. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, USA, 2,* 1709–1718.

Simon, H. A. (1973). The organization of complex systems. In H. H. Pattee (Ed.), *Hierarchy theory: the challenge of complex systems,* Chapter 1, pp. 1–28. New York: George Braziller.

Skinns, D. (1998) 'Crime reduction, diffusion and displacement: evaluating the effectiveness of CCTV', *Surveillance, Closed Circuit Television and Social Control.* Aldershot: Ashgate.

Smith, E. E., & Osherson, D. N. (1989) Similarity and Decision Making. *In Similarity and Analogical Reasoning, ed. S. Vosniadou and A. Ortony, UK, 60-75*

Smith, G. J. D. (1999). Behind the Screens: Examining Constructions of Deviance and Informal Practices among CCTV Control Room Operators in the UK. *Surveillance & Society, UK, 2(3),* 376-395

Solomon, K. O., Medin, D. L., & Lynch, E. L. (1999). Concepts Do More Than Categorize. *Trends in Cognitive Sciences, 3(3),* 99-105

Stauffer, C., & Grimson, W.(1999). Adaptive Background Mixture Models for Real-Time Tracking. *In Proceedings of IEEE Computer Vision and Pattern Recognition, 2,* 246-252

Stolcke, A. (1995). An Efficient Probabilistic Context-Free Parsing Algorithm that Computes Prefix Probabilities. *Journal of Computational Linguistics, 21(2),* 165-201

Thorpe, S. J., Delorme, A., & VanRullen, R. (2001). Spike Based Strategies for Rapid Processing. *Neural Networks, 14, 715-726*

Turaga, P., Chellappa, R., Subrahmanian, V. S., & Udrea, O. (2008). Machine Recognition of Human Activities: A Survey, *IEEE Transactionson Circuits and Systems for Video Technology, 18(11),* 1473–1488.

APPENDICES

APPENDIX A

Original Earley's Parser Implementation

```
function EARLEY-PARSE(words, grammar)
    ENQUEUE((γ → •S, 0), chart[0])
    for i ← from 0 to LENGTH(words) do
        for each state in chart[i] do
            if INCOMPLETE?(state) then
                if NEXT-CAT(state) is a nonterminal then
                    PREDICTOR(state, i, grammar     // non-terminal
                else do
                    SCANNER(state, i)                 // terminal
            else do
                COMPLETER(state, i)
        end
    end
    return chart

procedure PREDICTOR((A → α•B, i), j, grammar),
    for each (B → γ) in GRAMMAR-RULES-FOR(B, grammar) do
        ADD-TO-SET((B → •γ, j), chart[ j])
    end
procedure SCANNER((A → α•B, i), j),
    if B ⊂ PARTS-OF-SPEECH(word[j]) then
        ADD-TO-SET((B → word[j], i), chart[j + 1])
    end
procedure COMPLETER((B → γ•, j), k),
    for each (A → α•Bβ, i) in chart[j] do
        ADD-TO-SET((A → αB•β, i), chart[k])
    End
```

APPENDIX B

RANSAC Algorithm

The generic algorithm of RANSAC algorithm in pseudocode, works as follows:

```
input:
    data - a set of observations
    model - a model that can be fitted to data
    n - the minimum number of data required to fit the model
    k - the number of iterations performed by the algorithm
    t - a threshold value for determining when a datum fits a model
    d - the number of close data values required to assert that a
model fits well to data
output:
    best_model - model parameters which best fit the data (or nil if
no good model is found)
    best_consensus_set - data points from which this model has been
estimated
    best_error - the error of this model relative to the data

iterations := 0
best_model := nil
best_consensus_set := nil
best_error := infinity
while iterations < k
    maybe_inliers := n randomly selected values from data
    maybe_model := model parameters fitted to maybe_inliers
    consensus_set := maybe_inliers

    for every point in data not in maybe_inliers
        if point fits maybe_model with an error smaller than t
            add point to consensus_set
```

```
    if the number of elements in consensus_set is > d
        (this implies that we may have found a good model,
        now test how good it is)
        this_model := model parameters fitted to all points in
consensus_set
        this_error := a measure of how well this_model fits these
points
        if this_error < best_error
            (we have found a model which is better than any of the
previous ones,
            keep it until a better one is found)
            best_model := this_model
            best_consensus_set := consensus_set
            best_error := this_error

    increment iterations

return best_model, best_consensus_set, best_error
```

METU LIBRARY

TEZ FOTOKOPİ İZİN FORMU

ENSTİTÜ

Fen Bilimleri Enstitüsü ☐

Sosyal Bilimler Enstitüsü ☐

Uygulamalı Matematik Enstitüsü ☐

Enformatik Enstitüsü ☐

Deniz Bilimleri Enstitüsü ☐

YAZARIN

Soyadı : Büyüközcü.....
Adı : Demirhan.....
Bölümü : Bilişsel Bilimler.....

TEZİN ADI (İngilizce) : DISCRETIZED CATEGORIZATION OF HIGH LEVEL TRAFFIC ACTIVITES IN TUNNELS USING ATTRIBUTE GRAMMARS.....
.....
.....
.....
.....

TEZİN TÜRÜ : Yüksek Lisans ☐ Doktora ☐

1. Tezimin tamamı dünya çapında erişime açılsın ve kaynak gösterilmek şartıyla tezimin bir kısmı veya tamamının fotokopisi alınsın. ☐
2. Tezimin tamamı yalnızca Orta Doğu Teknik Üniversitesi kullancılarının erişimine açılsın. (Bu seçenekle tezinizin fotokopisi ya da elektronik kopyası Kütüphane aracılığı ile ODTÜ dışına dağıtılmayacaktır.) ☐
3. Tezim bir (1) yıl süreyle erişime kapalı olsun. (Bu seçenekle tezinizin fotokopisi ya da elektronik kopyası Kütüphane aracılığı ile ODTÜ dışına dağıtılmayacaktır.)
☐

Yazarın imzası ..... Tarih .....