Portable Traffic Data Processor

Final Report

Prepared by:

Nikoloas Papanikolopoulos
Harini Veeraraghavan

Department of Computer Science and Engineering
University of Minnesota

CTS 08-14

Technical Report Documentation Page

1. Report No.	2.	3. Recipients Accession No.	
CTS 08-14			
4. Title and Subtitle		5. Report Date	
Portable Traffic Data Processor		September 2008	
		6.	
7. Author(s)		8. Performing Organization Report No.	
Nikoloas Papanikolopoulos, Harini Veeraraghavan			
9. Performing Organization Name and Address		10. Project/Task/Work Unit No.	
Department of Computer Science and Engineering University of Minnesota 4-192 EE/CS Building 200 Union Street SE Minneapolis, MN 55455		CTS project # 2006033	
		11. Contract (C) or Grant (G) No.	
12. Sponsoring Organization Name and Address		13. Type of Report and Period Covered	
Intelligent Transportation Systems Institute University of Minnesota 200 Transportation and Safety Building 511 Washington Avenue SE Minneapolis, Minnesota 55455		Final Report	
		14. Sponsoring Agency Code	
15. Supplementary Notes			
http://www.cts.umn.edu/Publications/ResearchReports/			

16. Abstract (Limit: 200 words)

Automatic extraction of events from video sequences has important applications in a variety of Intelligent Transportation Systems (ITS) problems including scene monitoring, traffic data collection, intersection monitoring, etc. When deploying a system that recognizes events automatically from video sequences, two important things to consider are the real-time analysis of the video sequences and fast learning times required for learning the different classes of events in a scene. A related requirement which is often ignored is the limited reliance of the learning system on the user provided knowledge. In this work, we present an innovative technique for detecting the different events in video sequences through a semi-supervised learning method.

17. Document Analysis/Descriptors	18. Availability Statement		
Events, learning, tracking, computer vision	No restrictions. Document available from: National Technical Information Services, Springfield, Virginia 22161		
19. Security Class (this report)	20. Security Class (this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	47	

Portable Traffic Data Processor

Final Report

Prepared by:

Nikoloas Papanikolopoulos
Harini Veeraraghavan

Department of Computer Science and Engineering
University of Minnesota

September 2008

Published by:

Intelligent Transportation Systems Institute
Center for Transportation Studies
University of Minnesota
200 Transportation and Safety Building
511 Washington Ave SE
Minneapolis, Minnesota 55455

Contents

Tables

Figures

Executive Summary

Automatic extraction of events from video sequences has important applications in a variety of Intelligent Transportation Systems (ITS) problems including scene monitoring, traffic data collection, intersection monitoring, etc. When deploying a system that recognizes events automatically from video sequences, two important things to consider are the real-time analysis of the video sequences and fast learning times required for learning the different classes of events in a scene. A related requirement which is often ignored is the limited reliance of the learning system on the user provided knowledge. In this work, we present an innovative technique for detecting the different events in video sequences through a semi-supervised learning method. More concretely, the events are recognized from the tracked trajectories of the targets in the scene which in turn are represented as a collection of actions or strings. By parsing these strings as reversible context-free grammars we detect and classify the different events. Learning consists of extracting the relevant grammar for each class of events from the data. To accomplish learning goal, the system makes use of a small number of trajectories corresponding to each class as provided by an user to obtain a preliminary model of the grammar. Using this model, the system iteratively refines the grammar from new trajectory data obtained directly from the scene. Given that the system requires only a very small number of labeled trajectories and can iteratively learn from the observed data, the system is easily portable to new scenes with little system initialization from the user.

Chapter 1

Introduction

1.1 Introduction

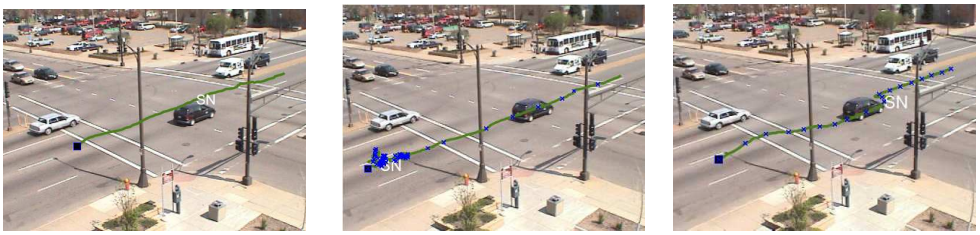
Automatic analysis of video sequences using computer vision techniques for event detection has several interesting applications in Intelligent Transportation Systems. However, the real-time computational constraints, the unconstrained nature of outdoor video sequences, and the limited availability of user provided knowledge make automatic event detection a very challenging task. Till now, work in recognizing dynamic events from video sequences has been addressed as some form of unsupervised clustering based on the analysis of a large number of examples such as [51, 61, 31], event classification using a trained model of hidden Markov models [4, 23, 41], as well as predefined models of context-free grammars [19, 34] and variations including [14]. It is already known that context-free grammars provide a flexible representation for the events and hence can model a diverse variations in a class of events in comparison to hidden Markov models (HMM) and not require an exorbitantly large training data set like the unsupervised methods using dimensionality reduction techniques. However, very minimal attempts have been made thus far in representing the events arising in video sequences as patterns of strings parsable using context-free grammar methods [19, 34]. Additionally, to the authors' knowledge no attempt has been made to learn the grammars for producing vision-based events. In this work, we present a novel entropy-minimization based criterion applied to an iterative best-first search based optimization method for learning the set of grammars for producing the different classes of events. An initial guess of the grammars are

learned using a small set, typically, 5 to 6 trajectories for each class labeled by the user.

The reminder of this chapter is organized as follows: We first describe the problem of dynamic event detection in video sequences in Section 1.2. We then introduce the relevant definitions such as the representation of the events as strings of actions, a context-free grammar, the parsing methodology, etc in Section 1.3. We then introduce the learning method and provide basic intuition for the learning method and the issues related to extracting the grammars in Section 1.5.

1.2 Problem Description

Dynamic events in video sequences typically arise from spatio-temporal trajectories with varying temporal scales. Temporal variations arise from the varying speeds in the motion of the targets, namely, vehicles in a scene. Additionally, vehicles exhibit a wide range of motion characteristics for the same type of event. For example, Fig. 1.1 illustrates three different examples for an event such as a vehicle moving through the intersection from south to north. The three motions consists of a stop-n-go motion, where a vehicle stops for a certain period of time before moving through the intersection, a continuous motion where a vehicle moves through the intersection with more or less the same speed, and finally, a lane changing motion where a vehicle executes a lane change during its motion through the intersection. These are just a small set of variations that vehicles depict for a same event. Thus, one of the requirements is that the event detection method be robust to the large number of variations in an event both from temporal variations as well as the structure of motion in a scene. Additionally, like any computer vision problem, robust event detection in general outdoor



(a) Typical South-North motion.

(b) Stop-n-go South-North motion.

(c) Lane change South-North motion.

Figure 1.1: Example categories of south-north through intersection motion. As shown, several sub-categories arise for the same event class.

scenes, relies heavily on the robust detection and tracking of the targets. Thus, the system needs to be robust to the

underlying scene clutter, varying illuminations and noisy segmentation resulting from the motion of the camera in the scene. We have addressed the problem of robust target registration and tracking with reasonable success in our earlier work [55]. The work described in this work assumes that the data for event classification, namely the target trajectories are obtained through a robust segmentation method. However, the obtained trajectory may contain missing data in the form of incomplete or partial trajectory segments as well as incorrectly labeled actions. The method for robust event classification is discussed in the succeeding Chapter 2.

1.3 Relevant Definitions

Data Representation as Strings of Actions The individual target trajectories are represented as motion strings. A motion string consists of a group of actions that were executed by the target along its path across the scene. An action typically consists of the spatial region occupied by the target in the scene along with the average motion executed by the target during a certain time interval. The time interval is predefined. An example motion string for a target is represented in Fig. 1.2. The corresponding target trajectory for the same string is depicted alongside. As shown, the image is discretized in to a number of arbitrary polygonal regions or cells. The spatial region labels are obtained from the region occupied by the target. As such, these regions can also be assigned semantically meaningful names such as, "intersection region", "stopping region", "turn lane", "carpool lane", etc, as appropriate to the application domain. Throughout this report, we assign arbitrary labels to the different regions such as $c_1, c_2, \ldots, c_n$, where n is the number of regions. The local motions are indicated as one of "straight moving", "stopping motion", "fast moving", "turning left", "turning right", etc. These motions are computed using samples of the target trajectory across a sliding window of fixed length. The actions are represented as a pair of local motion and the spatial region corresponding to the local motion. An example string is depicted in Fig. 2.2, where $C1, C5, C6$ correspond to the discrete spatial cells and (*straight, fast)*correspond to the local motion. The chosen representation of the target trajectory as motion strings allows us to obtain a succinct and meaningfully summarized description of the target's motion across the scene. This in turn allows us to easily parse and classify the event. However, the scale or resolution of the events that can be detected depends on the temporal window chosen for sampling the trajectory to compute the motion string.

(a) Target trajectory in Image

C1(straight) C1(straight) C1(straight) C5(fast) C5(fast) C6(straight) C6(straight) C6(straight) C6(straight)

(b) An example motion string

Figure 1.2: Example cell-based representation of the spatial region. The cells are shown as white regions with the corresponding labels. An example trajectory of a lane-changing vehicle is shown in black. The corresponding motion string for the same trajectory is shown alongside.

Context-Free Grammar for Parsing the Events A context-free grammar consists of a tuple (Ξ, N, T, Γ), where, Ξ is the language of the grammar consisting of a set of terminal symbols, such as the individual actions in the motion string, N is the set of non-terminals whose derivation yields a set of terminal symbols or another non-terminal, T is the starting non-terminal from which the grammar derivation is started, and Γ is a set of rules or productions for deriving the terminals or non-terminals in the grammar. Probabilistic version of the grammar referred to as Stochastic Context-Free Grammars (SCFG) or Probabilistic Context-Free Grammars consist of another parameter or a probability term associated with each rule in the grammar.

A context-free grammar allows us to derive the different symbols in a string with the assumption that the individual labels are generated independent of each other. This provides a huge flexibility in the derivation of the strings. An example grammar for a south to north through intersection event is depicted in Table. 1.1. Using this grammar, one can derive an event such as $C1(straight)C5(fast)C5(fast)C6(straight)C6(straight)C6(straight)$ with probability $0.2 \times 1.0 \times 1.0 \times 1.0 \times 0.4$. The context freeness is evident from the derivation of the string where the production of the subsequent symbol is independent of the preceding symbol. The individual probabilities

correspond to the probability of the production or the rule for generating the string. The details of the derivation

$T \rightarrow M1M2M3M4$	0.7
$T \rightarrow M2M3M4$	0.2
$T \rightarrow M3M4$	0.1
$M1 \rightarrow C1(straight)C1(straight)$	1.0
$M2 \rightarrow C1(straight)C5(fast)$	1.0
$M3 \rightarrow C5(fast)C6(straight)$	1.0
$M4 \rightarrow C6(straight)C6(straight)M4$	0.6
$M4 \rightarrow C6(straight)C6(straight)$	0.4

Table 1.1: Examples to chastic context-free grammar for generating a sub-class of south-north through intersection motions.

of the string using the standard Earley Parser is discussed in subsequent sections.

Semi-Supervised Learning Method The semi-supervised learning method combines the goodness of both the supervised and unsupervised learning methods. While supervised learning methods directly present the relevant data for a specific class, thereby, easing the learning problem for discovering the correct structure at the cost of having to label a large number of data, the unsupervised learning methods obviate the requirement for labeling a large amount of data which is unrealistic is most scenarios at the expense of large amount of data needed to discover the correct structure. The semi-supervised learning method on the other hand, tries to obtain a balance of the two learning methodologies by using a small amount of labeled examples for initializing the learner. Using the initial structure, the learner can then generalize from a larger set of unlabeled data.

1.4 Parsing the Events using Stochastic Context-Free Grammars

Before describing the parsing method, we introduce the basic notations and the definitions.

1.4.1 Notation and Basic Definitions

Following standard notation, the non-terminals are represented using upper case letters, while the terminals are represented using lower case letters. In this work, terminals consists of alphanumeric symbols as opposed to English alphabets alone. Strings of mixed terminal and non-terminal symbols are written using lower case Greek symbols, such as λ, κ, γ

etc., and empty symbol is referred to as ϵ.

The probability of a production $p(X \rightarrow \lambda)$ represents the probability with which the production resulting in the derivation of λ is chosen, whenever the non-terminal X is expanded. Hence, for a set of expansions, $\lambda_1, \ldots \lambda_n$ resulting from X, the probability of all such expansions sum to 1, yielding,

$$\sum_i p(X \rightarrow \lambda_i) = 1. \tag{1.1}$$

The probability of deriving a string x by a grammar M, $p(T \Rightarrow \ldots \kappa_k)$ where κ_k is combination of non-terminals and terminals, is given by

$$p(T \Rightarrow \ldots \Rightarrow \kappa_k) = p(T \Rightarrow \ldots \Rightarrow \kappa_{k-1}) p(K \rightarrow \lambda). \tag{1.2}$$

In general, a grammar model M can produce a string x through more than one sequence of derivations. In such a case, the probability of the string is obtained by taking the expectation of all possible derivations resulting in the string x. Thus,

$$p(T \Rightarrow \ldots \Rightarrow x) = \sum_{(T \Rightarrow \ldots \Rightarrow x)} p(T \Rightarrow \ldots \Rightarrow x). \tag{1.3}$$

In this work, grammars are assumed to be simple, which means that only a unique sequence of derivations exists for a particular string from a grammar.

Each class consists of a grammar G_i and is made up of a language $L(G)$ consisting of a set of terminals or alphabets represented by Ξ, a set of non-terminals N, a start symbol non-terminal T, a set of rules or productions Γ for deriving the non-terminals and terminals, and the probability of production $p(r), \forall r \in \Gamma$. A parse tree generated by a grammar G_i is indicated by c_i.

1.4.2 Earley Parser

An Earley parser makes use of the concept of states for matching an input string $s_1, s_2, \ldots, s_n$ with a grammar. A state represents: the production currently being matched with the input string, a pointer usually indicated by a '.' operator that indicates how much of the right hand side of the production has been expanded, and a pointer to the input string that represents how much of the input string has been scanned. A state can be represented as,

$$i : \ {}_kX \rightarrow \lambda.\mu$$

where i and k correspond to pointers to the point of expansion of the derivation tree and the input string. X is a non-terminal while λ and μ are terminals or non-terminals. The '.' to the right of λ indicates the point up to which the non-terminals in the production have been expanded. Probabilistic version of the parser consists of an additional probability term attached to each production:

$$X \rightarrow \lambda[p] \tag{1.4}$$

where p is the probability of the production.

An Earley parser consists of three basic operations, namely, prediction, scanning, and completion.

Prediction The prediction step of the parser expands the non-terminals on the right hand side of the dot, to generate different alternative derivations of the state. For instance, $i :\rightarrow \lambda.Y\mu$ is expanded to the set of alternatives $Y \rightarrow \nu[p]$ such that, $i :_i Y \rightarrow .\nu[p]$

Scanning The input string is matched with the terminals produced by left-most derivations from the prediction step. As a result of this step, the input pointer is advanced to the next input terminal upon the detection of a matching derivation.

Completion This step consists of updating the markers of all the pending derivations, $j : \ {}_kX \rightarrow \lambda.Y\mu$ given the detection of complete states [1], $i : \ {}_jY \rightarrow \nu.$ where $j \leq i$. Complete states are obtained as a result of prediction and the matching of the input symbols in the scanning step.

In the probabilistic setting, the parser consists of two sets of probabilities, called the forward and inner probabilities,

[1] completeness is indicated by the dot shifted to the right of the derivation.

similar to the forward and backward probabilities in the Baum-Welch algorithm for HMM estimation. These probabilities are used for pruning and selecting a set of states for derivation. The effect of these probabilities is to constrain the derivations of the parse trees such that only derivations consistent with the forward probabilities are allowed to expand. The forward probability, also called the prefix probability is the probability of the parsed string up to a point i and the inner probability is the probability of a sub-string from position i through k. In the extension introduced by Stolcke, an additional probability called the Viterbi probability is introduced in the derivations, which is used to trace through the paths of the derivations, in order to pick the most likely derivation path for generating the input string.

Both Earley and Stolcke's extension to the Earley parser assumed the input to be free of noise. The problem of noisy input is addressed by [19] by the inclusion of the probability of the input in the computation of the forward and inner probabilities:

$$\alpha_i = \alpha_{i-1}(i :_k X \rightarrow \lambda.a\mu)p(a) \tag{1.5}$$

$$\gamma_i = \gamma_{i-1}(i :_k X \rightarrow \lambda.a\mu)p(a) \tag{1.6}$$

where α and γ are the forward and inner probabilities, and $p(a)$ is the probability of the input. Hence, this provides a nice way of mapping the uncertainty in the input to the derivation of the parse tree.

Another problem associated with parsing is error recovery on the encounter of unmatched strings. Moore and Essa [34] propose error recovery for insertion, substitution, and deletion errors. Insertion errors occur as a result of insertion of spurious symbols in the input string, while substitution errors occur due to the replacement of a symbol by another. Thirdly, deletion errors occur due to the missed detection of a symbol. Their solution for insertion errors simply consists of ignoring the symbol and moving to the next symbol. The presence of other two errors results in maintaining multiple derivation paths for all the derivations that can result for the current string position.

1.5 Learning Event Classifications

Learning for event classification typically consists of solving two different problems:

1. **Learning the structure of model.** In a stochastic context free grammar, this essentially consists of learning the set of non-terminals, terminals, and productions or the rules.

2. **Learning the parameters of the model.** This typically consists of estimating the continuous parameters of the model or the probabilities associated with the individual rules.

Problem (**a**) is the difficult one of the two learning problems. This problem can be solved by specifying a grammar corpus and applying parameter estimation to learn the model parameters or solve problem (**b**). The rules with very low probabilities are then pruned as irrelevant to the grammar. However, this requires providing a sufficiently large grammar corpus. In this work, we take a different approach, wherein, we incrementally, obtain the grammar by iteratively learning from the example data.

1.5.1 Estimating Continuous Model Parameters

Given that this is the easier of the two learning problems, most solutions to learning events resort to parameter estimation by specifying a very general structure of the model. Typically maximum likelihood-based approaches including the forward-backward or the Baum-Welch algorithm, gradient descent-based methods, etc., are commonly used for HMMs and the inside-outside algorithm [27] for the SCFGs. Both of these approaches are based on the Expectation Maximization (EM) algorithm where the continuous parameters are refined iteratively in each step by maximizing the likelihood of the model. Starting from an overtly general model and refining solely the continuous parameters leads to an unconstrained optimization of the structure. The only disadvantage of an Expectation Maximization style search algorithm is the danger of getting stuck in local minima, and the strong dependence of the solution on the initialization conditions.

1.5.2 Learning Discrete Model Structure

As mentioned earlier, this is the harder of the two problems. In case of SCFGs, one common approach is to build a tree-like data structure from the input data which exactly produces all the input strings and then refines the structure in order to generalize to new strings. Stolcke [52] introduced a model-merging based method for learning new rules from the data. The basic idea of this method was to construct complex models of the world by merging simple sub-models obtained from the data.

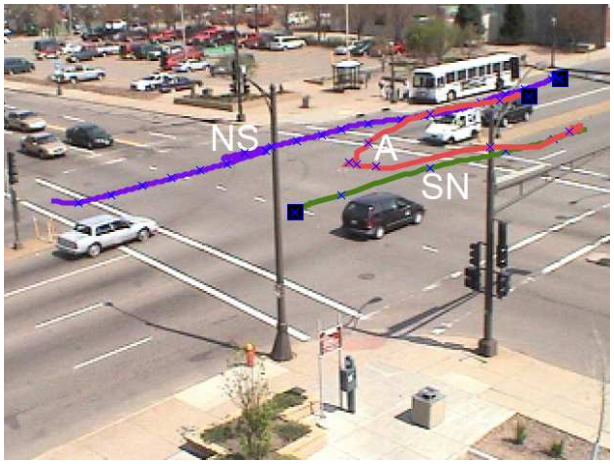


Figure 1.3: Classification for overlapping classes.

1.6 Issues in Learning

Two of the most important concerns associated with learning are:

- Limited size of training data, and
- Noise in the data.

Learning under the above two mentioned conditions is one of the main contributions of this thesis. In most environments including outdoor image sequences, obtaining a large amount of labeled data is unrealistic, although it is relatively easy to get a large amount of unlabeled data. In addition, noise in the training set is unavoidable due to the ambiguity of inference from the underlying vision data. Another concern with respect to learning classifications for multiple classes is the amount of overlap between the different classes. In increasingly complex environments, the amount of overlap between classes can be large. For example, consider the trajectories of vehicles executing a north-south and U-turn motion from north. At-least half the length of trajectories overlap between these two classes. In such cases, it is difficult to devise a learning algorithm that could correctly learn the different class descriptions. As shown in the Fig. 1.3, this work addresses these problems to attain robust classification.

1.7 Proposed Solution: Learning from Increasingly Complex Examples

Learning from positive examples alone is a difficult problem as the learner has to overcome both over-generalization and over-specialization or over-fitting. It has been shown in previous work [7], that this problem can be made more tractable when the learner learns from simple examples x specifying the language L being modeled. In other words, it was shown that by minimizing the Kolmogorov complexity of the examples, the learning problem can be made more tractable and can be guaranteed never to over-generalize. Entropy is directly related to Kolmogorov complexity. Thus, using the entropy in a multi-class learning problem, simplest examples can be attributed as those having the least classification entropy. This typically results when the example input string can be uniquely attributed to be produced by utmost one or no class. Considering only the cases where there is at least one class associated to an example, entropy provides the clearest way of quantifying the relation between an example and a class.

The iterative learning procedure essentially performs a best-first search, where the examples with the least classification entropy are used for updating the grammars of the different classes. The search stops when the total conditional classification entropy is reduced below a predefined threshold, or when no improvement beyond a particular threshold can be made by adding any additional examples, or when no more examples are left to augment the grammars. The conditional entropy of a motion string pattern is defined as,

$$H(y|s) = -\sum i = 1^{K} \sum j = 1^{M} p(y|s(j), \Omega_i) log(p(y|s(j), \Omega_i)) \tag{1.7}$$

where $H(y|s)$ is the conditional entropy of the string pattern $s = s_1, s_2, \ldots, s_M$, K are the set of all the grammars for K event classes, and M is the length of the string pattern. Ω_i is the grammar for event class i. The total conditional classification entropy consists of an additional summation term over all the examples, and is represented as,

$$H(Y|S) = -\sum t = 1^{N} \sum i = 1^{K} \sum j = 1^{M_i} p(y(i)|S(i,j), \Omega_i) log(p(y(i)|S(i,j), \Omega_i)) \tag{1.8}$$

where $Y = y_1, y_2, \ldots, y_N$ is the set of labels for N examples, and $S = s(1), s(2), \ldots, s(N)$ are the N example patterns.

In this work, we assume that the class of grammars used for describing the events are structurally reversible. A struc-

turally reversible grammar is one of type where among all the non-terminals that might derive a given string, no one is an extension of the other. In other words, when $A \rightarrow \alpha$ and $B \rightarrow \alpha$, then $A = B$. Again, when $A \rightarrow \alpha B \beta$ and $B \rightarrow \alpha C \beta$, then $B = C$. Here, the capital letters correspond to the non-terminals and the Greek letters indicate the terminal symbols.

1.8 Conclusion

This chapter presented the basic approach to dynamic event recognition in video sequences and discussed the issues and challenges involved in the same problem. The basic intuition and the semi-supervised learning approach along with a novel representation scheme for dynamic video-based events was outlined. This chapter also introduced the basics of parsing strings of actions as context-free grammar using a standard approach, namely, the Earley parser.

Chapter 2

Event Detection in Outdoor Video Sequences

Overview

Automatic detection of dynamic events from video sequences with limited reliance on user-provided knowledge is a challenging task for most computer vision based applications with a variety of applications including video segmentation and summarizing, visual surveillance and monitoring, intelligent transportation systems, highlight extraction, key-frame detection, and many more. As such, the events that arise from the motion of targets in a scene consist of varying temporal scales due to varying speeds of the target motion. In this work, we present a novel semi-supervised learning approach to learn the description of the different classes of events in a domain. The events are represented as patterns of motion strings based on the actions executed by the targets while they were visible in the scene. The said motion strings are then parsed using context-free grammars for event classification. The learning approach makes use of an entropy minimization regularization-based heuristic to guide an iterative search algorithm for learning the grammar for each event class.

2.1 Introduction

Dynamic events in image sequences result from spatio-temporal trajectories with varying temporal scales. Temporal variations can result from varying speeds in motion (such as motion of a vehicle) or in the execution of a task by an agent (such as a robot picking an object). Automatic extraction of such events from image sequences with limited knowledge or models is a challenging problem. Existing approaches to event detection can be broadly classified based on (i) the models

used for the detection as parametric [22, 8] and non-parametric or statistical methods [4, 23, 14, 19, 34], (ii) the goal of classification; to detect abnormal occurrences [61, 21] or to detect a predefined set of actions [46, 58, 19, 34, 23, 41]. The models for event recognition are usually exactly specified in the case of parametric approaches while non-parametric or statistical approaches either refine a pre-specified model structure from the observations or just estimate the continuous parameters. The main disadvantage of supervised methods is the requirement of a large labeled training set which is not realistic to obtain in most generic environments. Unsupervised methods such as [31, 35] obviate the need for labeled data but still require a larger training set and are limited in their capability as the complexity of the events or the overlap between the different event classes increases. Another concern for learning in realistic scenes is the noise in the input data. Thus the structure inferred by methods such as in [9, 21, 45] may be adversely affected by the noise in the data. Similarly, given the wide variations of the data, standard hidden Markov model approaches are ineffective and require more complex formulations such as hidden semi-Markov models [16, 40, 37] whose training can also be made intractable given the limited size of the training data and the noise.

In summary, detection of dynamic events in real-world image sequences should be able to:

1. Provide robust classification despite the broad variations in the occurrence of activities for one or more classes,
2. Accurately learn the description of activities with little supervision, and
3. Accurately learn as well as classify data corrupted by noise, or missing data. For example, in image sequences, localization of a target can be lost temporarily owing to occlusions.

This work addresses the problem of learning and the detection of typical as well as atypical dynamic events in realistic outdoor image sequences. A semi-supervised learning approach is used for extracting the set of events in a given image sequence from limited labeled data using stochastic context-free grammar. Instead of the commonly used approach consisting of specializing from a corpus and formulating the learning as a parameter estimation problem in the generative learner setting, this work makes use of a discriminative learner where the grammar structure consisting of the set of terminal symbols or the individual actions occurring in the string, the non-terminals that can derive terminal symbols or other non-terminals, productions or rules for deriving the motion strings are learned from the data. Following the

extraction of the grammar, their parameters,namely, the production probabilities are re-estimated.

The learning approach consists of presenting the system with a small set of labeled examples. In other words, the examples along with the classes that they correspond to are presented, using which, the system automatically learns a preliminary model of the grammar for deriving the motion strings for each class. Following this learning, the system is presented with an unlabeled data-set consisting of the motion strings without their class labels. The system incrementally refines the grammar for each event class by selecting the examples with the least classification entropy. Classification entropy corresponds to the extent of confusion in assigning an example uniquely to a particular class. Hence, by making use of examples that have the least confusion, the system is capable of learning an appropriate grammar for each event class.

2.1.1 Interesting Events Definition

What constitutes an interesting event is very application-dependent. In this work, event detection is motivated by the problem of automatic data collection in traffic intersections. Therefore, the events of interest consist of typical events such as left turning motions along a particular direction such as a south-west left turn, as well as atypical events including, U-turns, vehicles moving on unusual motion paths, etc. The main challenge in detecting such events arises from the noise in the underlying image sequences, variations in the data for a given event such as due to varying temporal scales, and limited number of labeled data sequences.

2.1.2 Chapter Outline

This chapter is organized as follows: The learning problem is formally stated in Section 2.2. We then present the related work in Section 2.3, followed by a brief background 2.4. The learning method is discussed in detail in Section 2.5.2,

2.2 Problem Statement

Given a spatio-temporal pattern S, expressed as a string of actions $S = \{a_1, a_2, \ldots, a_n\}$, with $1, \ldots, n$ being the discrete-time sampling intervals, we seek the grammar G_i corresponding to an event class E_i that can generate the said

pattern.

Given a fully specified SCFG, that is with fixed non-terminals and terminals, the inside-outside algorithm [27] is the optimization algorithm for estimating the parameters or rule probabilities. However, automatically learning the complete structure, namely, the terminal symbols, the non-terminals, the productions from data is generally a much difficult problem. This is because, it is possible to produce several grammars that could generate the same string by different combinations of the terminals and the non-terminals. However, when some clue of the valid combinations are available in the form of bucketing information, and when the class of grammars used for the task are assumed to be structurally reversible, the problem is much simplified. A structurally reversible grammar is one where no one non-terminal can derive a string of terminals and non-terminals such that one is an extension of the other. In other words in a structurally reversible grammar, when $A \rightarrow \alpha$ and $B \rightarrow \alpha$, then $A = B$. A, and B are the non-terminals, and α is a terminal symbol. Similarly, when $A \rightarrow \alpha C \beta$, and $A \rightarrow \alpha D \beta$, then $C = D$.

In most applications where such grammars are applied, the data is assumed to contain no noise. However, in the case of computer vision-based applications, the data obtained through tracking contains ambiguity and errors such as missing sub-sequences due to occlusions, as well as incomplete or partial sequences due to mis-tracking. Learning a robust grammar under these settings makes the problem even more challenging.

2.3 Related Works

Existing work in event detection can be broadly classified based on the methods used for detection as parametric model or rule-based or non-parametric or statistical learning.

Model or rule-based methods may consist of explicit geometric or appearance models of the target either specified ahead of time or learned through supervision, as well as constraints specified in the form of rules. For example, specific models of geometric appearance as in [22, 8] can be used for recognizing the actions of a target. Other approaches make use of dimensionality reduction methods to infer the hidden structure of appearance or motion of the targets such as in [9, 58, 33]. Temporal context based on periodicity of motion inferred from the data using supervised or unsupervised clustering methods such as in [46], temporal trajectory of motion or motion history images [2] are other examples of

model-based approaches.

Context or rule-based approaches make use of rules or constraints specified by the user for detecting activities. Rules can be based on spatial contexts [32, 26], temporal or motion context derived from image motion such as in [13, 44, 24] as well as specific motion dynamics, interaction constraints [47, 50]. A combination of different cues such as image motion and time [24] has been used to compute robust features for event detection to deal with noise in the video sequences. Rule-based constraints can be used either for segmenting the videos such as [13] which are then used as features for identifying the scenario or for directly categorizing the activity [44]. Contexts or constraints can also encode higher level knowledge for event detection such as in [50]. However, both model-based and rule-based approaches are limited in the number of activities that can be recognized or at-least require an extensive design and specification of the different rules by the user. Dimensionality reduction based methods which infer the hidden structure in the data such as [58, 33] require the data to be labeled and the training data to be free of the effects of noise so as to infer the appropriate structure. As mentioned earlier in the earlier Section 2.1, it is not easy to obtain a large supervised noise-free training data in most real-world environments.

Non-parametric or statistical approaches obtain either a statistical model of the events from observations or infer the hidden structure of the events through dimensionality reduction. Manifold learning-based approaches such as in [39, 45] obtain the structure of various events by reducing the dimensionality of the video sequences.

Events are more challenging to detect when the same event has variations in appearance, temporal scale, etc. Stauffer and Grimson [51] built a "code-book" to represent the variations of the same or similar targets. Similar to this approach, Zhong *et al.* [61] proposed an unsupervised learning method based on dimensionality reduction. A co-occurrence matrix is computed from the set of observed video sequences which is then used for recognizing unusual occurrences in video sequences. Manor and Irani [31] introduced a self-tuning spectral clustering approach for detecting events from video sequences by using temporal contexts across multiple temporal scales. Johnson and Hogg [21] employ an unsupervised method for learning the statistical distribution of target trajectories in a scene in order to detect abnormal motions in human surveillance videos. Naphade and Huang [35] introduced an unsupervised clustering approach using a fixed number of ergodic hidden Markov models for detecting and classifying video sequences. Unsupervised learning approaches have the advantage that the labels of the various classes of activities need not be specified. However, these methods are still

not robust to noise in the data since all data is weighed equally. Hence, in the presence of noise, incorrect structure inference may result. Other approaches for event detection include, anti-face method [42] and a habituation based approach developed by Itti and Baldi *et al.* [18]. In this work, events that occur frequently are learned as normal through an exponential decay function weighting the relative novelty of the event.

Hidden Markov models are most commonly used for event detection from time-series sequences. Approaches vary from completely designing the HMM to learning the continuous parameters of the HMM to prune its structure. For example, Kamijo *et al.* [23] applied a standard hidden Markov model (HMM) for detecting interesting events in traffic scenes resulting from relative spatial proximity of vehicles such as near-misses, passes, collisions, etc. Maximum entropy-based learning methods were used by the work in [11, 28, 30] for detecting and classifying various activities in video. Entropy minimization-based learning was introduced by Brand and Kettnaker [3] for pruning the structure of HMM for learning the typical pattern of activities in video sequences. Learning in all these cases is supervised henceforth, requiring a large set of labeled data.

Standard hidden Markov model formulation is inappropriate for detecting dynamic events when there is large variations in the data as well as discontinuities are present. Hence, more complex formulations of HMMs such as mixture HMMs, layered HMMs, hidden semi-Markov models etc., are used for activity recognition. Li and Porikli [29] used a mixture of Gaussian-based HMMs with video features extracted by applying discrete cosine transform to detect interesting traffic patterns. The events of interest are based purely on the density of traffic flow such as heavy congestion, light traffic conditions, etc. Gong and Xiang [10] used a dynamically linked hidden Markov model for learning the temporal relationships of multiple targets in a scene for learning the group interaction behaviors. The model is learned from the data using a Schwarz's Bayesian information criterion. Hongeng and Nevatia [16] used hidden semi-Markov models for recognizing events from video sequences. Similar to our approach, shape and motion of the targets are used to build a set of primitive events. The collection of primitive events form composite events which are then classified using a semi-hidden Markov model. Unlike our approach where the primitive events are directly detected based on the target motion and spatial location, the work in [16] employs a Bayesian network for classifying these events. Layered and hierarchical versions of HMM have been recently proposed in the work of [40, 41, 37] for event detection from video sequences for individual as well group behavior. Other examples include [4] for dealing with target interactions, [36] for non-exponential duration of

events, and a hierarchical HMM to combine bottom-up image information with the top-down context information [57]. However, as the complexity of the model increases so does the complexity of learning the model and in cases where the data size is limited, there is a danger of over-fitting the model resulting in poor generalization to unseen examples.

Semi-supervised learning combines the advantages of both supervised and unsupervised learning by bootstrapping the model with a small set of labeled examples and refining the model using unlabeled examples. Recently, Zhong *et al.* [60] used a semi-adapted hidden Markov model for learning a fixed number of unusual events in the scene. Learning is initiated using a model of usual events learned from the data. Unusual events are detected based on the low probability segments occurring in the new data which is then used to seed a new hidden Markov model for learning the unusual event. This approach however, can result in over-fit models for unusual events given the small size of the data-set used to learn such models. Nigam *et al.* [38] used a semi-supervised approach consisting of weighted EM and naive Bayes for text classification. Recently, Grandvalet and Bengio [12] proposed an entropy-minimization based regularization for semi-supervised learning of manifolds.

Context free grammars are other means of detecting events. These approaches can easily deal with the variations in the data as opposed to requiring complex model formulations of hidden Markov models. Hamid *et al.* [15] recently proposed an approach to detect anomalous activities from video sequences using tri-grams where the individual sub-sequences or sub-classes of typical events are hand-coded by the user. Unusual activities are then detected based on any observed atypical ordering or non-typical combination of the known sub-classes of activities. Hakeem and Shah [14] used a graphical network with hand-coded grammar for detecting activities arising from target interactions in video sequences. Other examples of context-free grammars applied to detecting events in video sequences include [19, 34, 54]. Ivanov and Bobick applied stochastic context free grammars [19] for gesture recognition as well as for detecting some simple events in outdoor scenes involving a single target. In order to deal with noise, the probability of each symbol (obtained from a hidden Markov model) was incorporated in the forward and the inner probabilities of the parsing algorithm. An approach for dealing with noise in image sequences was introduced by Moore and Essa [34] using addition, deletion, and insertion operations similar to those used in edit distances for matching strings. The events detected were for a fixed number of entities in a scene. Wada and Matsuyama [56] used non-deterministic finite state automaton for selecting focus of attention in images for recognizing events arising from multiple entities in a scene. However, the set of be-

haviors dealt with in this approach consists of segmenting people as they walk in an indoor environment. Until now all video-based event detection approaches employing variants of probabilistic grammars do not learn the grammar structure or its continuous parameters from the image data owing to the amount of noise present in the input. Event detection is solely restricted to recognition or classification of activities based on a pre-specified grammar. This work addresses the problem of learning the grammar structure from the image data using a semi-supervised learning approach. Entropy minimization based regularization is used to select the order of learning from the examples thereby, allowing us to learn from simpler examples first before learning from the complex ones.

Stochastic context free grammar-based approaches have been used extensively in language modeling [20, 52], as well as in bioinformatics [49]. A good review of context free grammar-based parsing is in [1]. Learning of stochastic context free grammars (SCFG) consists of (i) estimating the topology and (ii) values of continuous parameters or the production probabilities from the data. Learning of the context free grammars especially as tree-based grammars has been studied by [25, 52, 59]. The main problem with the learning arises in the absence of any knowledge of the topology, and the terminals. Most of the current approaches to learning grammars consist of obtaining a more specific and concise representation starting from a grammar corpus such as the evolutionary approach used by [59, 25]. Extensive research exists for estimating the continuous parameters such as the inside-outside algorithm [27] and Bayesian model-merging approach [53]. Collins [6] presents a good review on the theory of statistical parameter estimation methods for context free grammars. A polynomial time algorithm for learning grammar topology with wholly positive examples and partially known structure of data was proposed by Sakakibara [48]. Parekh and Honavar [43] present an extensive review on grammar learning for natural languages using deterministic finite automata, context-free grammars, stochastic context free grammars, etc.

One important issue with learning is the absence of labeled examples. In most domains it is not possible to obtain a large corpus of labeled data. Semi-supervised learning is the most commonly used approach for such problems. When using semi-supervised learning, care needs to be taken on how the unlabeled data are utilized for learning the model [5]. Nigam *et al.* [38] used a semi-supervised approach consisting of weighted EM and naive Bayes for text classification. The basic idea of using naive Bayes classifier is that a small set of labeled data is employed to construct the initial data distribution. Bayesian classification is then applied on the remaining unlabeled data to produce classification, which in

turn iteratively estimates the distribution until convergence. The danger with this approach is that any early association of data to a particular class may result in incorrect classification. Inoue and Ueda [17] proposed extended tied HMMs for training with labeled and unlabeled time series data. The algorithm makes use of an iterative Baum-Welch style algorithm. The labeled data is used to bootstrap the individual tied-HMMs which are then tied over the states as well as classes for estimation using unlabeled data. In essence, the unlabeled data are assumed to arise from a mixture of classes, with the weights corresponding to each class re-estimated in the EM algorithm. One advantage of [17] is that it models the dependencies of the data in a time series.

2.4 Background

A stochastic context-free grammar consists of the following: (i) a set of terminals or alphabets λ, κ etc., of the language Ξ, a set of non-terminals N, a start symbol non-terminal T, rule or productions Γ for deriving the string of terminals and non-terminals, and the probability of each production in the rule set. The terminals correspond to the individual actions that an agent can perform in the given scene. The rules are used for deriving an event pattern obtained from a target trajectory. Given the grammar structure, the goal of the learning algorithm is to induce the set of relevant productions for generating the patterns corresponding to a particular event using a small set of labeled examples and a larger collection of unlabeled examples.

2.4.1 Learning From Increasingly Complex Examples

Learning using solely positive examples is in general a difficult problem as the learner has to overcome both over-generalization and over-specialization. It has been shown in previous work [7] that this problem can be made more tractable by learning first from examples with lower Kolmogorov complexity. In other words, the approach is to learn from simple examples before learning from the complex ones. The learning approach in this work is motivated by this idea.

The Kolmogorov complexity is directly related to the entropy. The entropy in the classification of a pattern corresponds to the uncertainty in attributing its membership to the k classes. Thus, lower entropy will essentially correspond to the case where the pattern has membership to fewer classes, thereby containing less ambiguity. In turn, such samples make

good candidates for training in comparison to those possessing large ambiguities. Thus, the event classes are updated using the examples that have unique membership first before using the examples that have multi-class membership, thereby preventing collapsing of the event grammars. The conditional classification entropy is expressed as,

$$H(y|s) = -\sum_{i=1}^{k}\sum_{j=1}^{M} p(y|s_j, G_k) log(p(y|S_j; G_k)) \tag{2.1}$$

where y is the output label $[0-1]$, $s = s_1, s_2, \ldots, s_M$ is the input spatio-temporal pattern, G_k is the grammar for class k. The spatio-temporal pattern consists or the motion-string consists of a collection of actions depicted by the target while it was visible in the scene.

To summarize, the basic approach to learning consists of iteratively refining the event grammars using the unlabeled examples which produce the lowest entropy. The individual class grammars are initially trained using a small number of labeled examples. Training consists of extracting a grammar structure. Another measure applicable for the stopping condition is the empirical conditional classification entropy, expressed as,

$$H(Y) = \sum_{i=1}^{N} H(y_i|S_i) \tag{2.2}$$

where there are N examples. The empirical conditional classification entropy is used as a measure of the convergence of the learning algorithm to the desired grammars. In other words, as learning progresses, the empirical conditional classification entropy should decrease. By using a threshold for the same, we can obtain a stopping condition for the learning problem.

2.5 Event Detection Using Context-Free Grammars

We summarize the event representation scheme and the idea of representing a target trajectory as motion string as discussed in the previous chapter, followed by the description of the learning algorithm.

2.5.1 Pattern Representation

An action sequence or a pattern is represented as a discrete set of primitive actions obtained by sampling from a target's trajectory. The sampling intervals are fixed beforehand. A primitive action is composed of the spatial location and the current local motion of the target obtained from an estimator such as a Kalman filter.[1] The local motions are are discretized into one of "straight moving", "stopped or slow moving", "fast moving", "left" and "right turning" through thresholding of the local velocity estimates and simple heuristics for turn detection. The spatial location is again obtained from the region occupied by the target in the image. For this purpose, the image is discretized into an arbitrary number of cells as shown in Fig. 2.1. The cells can either be laid out by the user or randomly generated.

The actions are represented as a pair of local motion and the spatial region corresponding to the local motion. An exam-

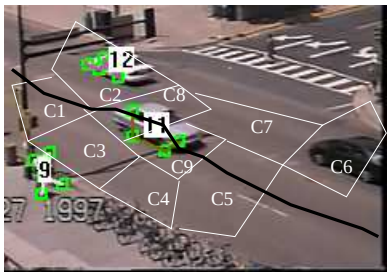


Figure 2.1: Example cell-based representation of the spatial region. The cells are shown as white regions with the corresponding labels. An example trajectory of a lane-changing vehicle is shown in black.

ple string is depicted in Fig. 2.2, where $C1, C5, C6$ correspond to the discrete spatial cells and (*straight, fast)*correspond to the local motion.

2.5.2 Learning Event Grammars

The set of events in the scene are assumed to arise from k different classes. In general, the same underlying distribution $D(s, y)$ is assumed to produce both the test and training examples. Learning consists of estimating the conditional distribution $D(y|s, \Omega)$ where y is the output for the input s, and Ω is the model that maps inputs to the outputs in a discriminative learning setting. The learning problem can be formulated as maximizing the conditional likelihood of the

[1] In our case, we use an extended switching Kalman filter for tracking the targets in the scene.

output label y given the input string s and classes $1, \ldots, k$ as,

$$\Delta = \sum_{i=1}^{N} \prod_{j=1}^{k} \sum_{l=1}^{M_i} p(y(i)|s(i,l);\Omega_j). \tag{2.3}$$

Entropy is the clearest way of characterizing the uncertainty in the posterior probabilities of the classification labels y for an input string s. In other words, an entropy zero corresponds to perfect classification, or the case when utmost one class is attributable to the given input string. With no need to invoke the independence assumptions, entropy provides a convenient way to express the relative certainty in attributing a pattern to the different classes. Expressed in terms of empirical conditional entropy, and assuming uniform prior on all the classes $1, \ldots, k$, Eqn. (2.3) becomes,

$$H(y|S) = \sum_{i=1}^{N} \sum_{j=1}^{k} \sum_{l=1}^{M_i} p(y(i)|s(i,l);\Omega_j) \log \frac{1}{p(y(i)|s(i,l);\Omega_j)}. \tag{2.4}$$

where $S = s_1, \ldots, s_N$ is the set of patterns. Again, the empirical conditional entropy is the lowest when the co-dependence of the label y and pattern s is high. The algorithm thus consists of refining the class grammars iteratively by selecting the examples that have the lowest entropy such that the empirical conditional entropy is minimized in each step. Using the examples with the least entropy essentially corresponds to performing a best-first search through the examples. The algorithm halts refinement when no more examples below a specified entropy level are left for update, or when the empirical conditional entropy is reduced below a threshold E_th, or when no improvement in the empirical conditional entropy can be obtained by the addition of any more examples. The basic algorithm is summarized in Table 2.1.

As depicted in Table 2.1, after learning a preliminary model with a few labeled training examples, the CFG model for each class if applicable in the given step is learned from the examples. The rule probabilities of the SCFG are then recomputed using the histogram counts of the number of times a rule was applied to classify the examples in the training set.

At the end of the iterative learning, the parameters, namely, the production probabilities can be reestimated using the standard optimization methods such as the inside-outside algorithm [27].

As mentioned earlier, the grouping of actions or the terminals for forming the non-terminals is assumed to be available

before-hand in the form of bracketed expressions. An example is depicted in Fig. 2.2. As shown, non-terminals are created from the terminal pairs (individual actions represented in the brackets) which are then merged with the newly created or previously existing non-terminals in the grammar. This helps to obtain a concise description of the rules. This process is similar to the non-terminal merging operation proposed by Stolcke [53].

The only prior knowledge the learning algorithm requires is some knowledge of the bracketing structure of the examples and a small set of supervised examples. In most real-world domains such as traffic intersection monitoring and human activity recognition, it is impossible to obtain a large amount of supervised learning examples as well as specify the structure of the scene. The proposed learning algorithm can easily be applied to data arising from the said applications with minimal user provided knowledge.

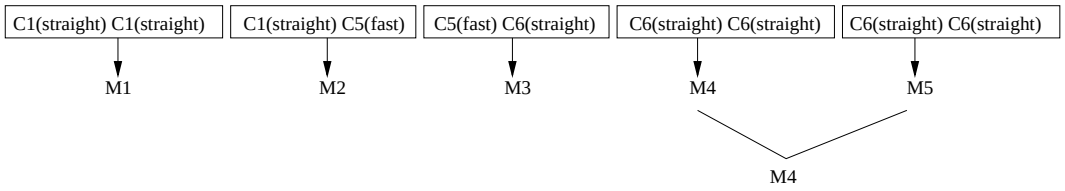


Figure 2.2: An example action sequence with braketed structure and non-terminal creation. The boxes around the strings represent the braketed structure. The terminals consist of the spatial cell occupied by the target such as C1, and the corresponding local motion, such as straight, fast, slow, left, right. The non-terminals are depicted as M1, M2, M3, etc.

```
 SUPERVISED: for each class k
   for each labeled string s belongs to k
      update grammar of k
   end
   UNSUPERVISED:
   do
    for each string s in 1 to N
      for each grammar j in 1 to K
        compute p(y|s,grammar(j))
//y=1 when classified into j, y=0 otherwise
      end
      //Compute conditional classification entropy
        E(s) = H(y|s, class1)+...+H(y|s, classk)
      end
    //compute empirical conditional entropy
      oldE=sum(E(s=1:M))
    [min_string_entropy, minIndex] = (minimum(E(s=1:M) > 0)
    do
    numUpdatedGrammars = 0
    if (min_string_entropy > entropy_threshold)
       then exit
    else
       for each grammar j 1 to N
          if(y==1 for string(minIndex) & grammar(j)) then
            update grammar j with string(minIndex)
    //recompute empirical conditional entropy
    newE = sum(E(i=1:M))
            if(newE > oldE)
              accept new grammar for class j
              numUpdatedGrammars = numUpdatedGrammars + 1
            end if
       end for
       min_string_entropy = minimum(E(1:M) > min_entropy)
    end else
    while (numUpdatedGrammars > 0)
  (while maximum trials | change_in_entropy < t)
```

Table 2.1: Algorithm for grammar update. The grammar for the typical classes is updated incrementally using the strings with the least entropy.

Chapter 3

Experimental Results

Overview

The objective of the experiments was to test the efficacy of the proposed learning method on real-world image sequences. Various sites were used and this chapter outlines our findings.

3.1 Experiment Description

For the experiments we chose scenes from outdoor traffic intersections such as depicted in the Fig. 3.1. Traffic intersections are one of the most complex scenes both for target localization as well as event detection. The uncontrolled environmental effects such as occlusions and changing illumination make target localization a challenging task. The resulting ambiguity in the observed data in addition to the significant overlaps between various events make event recognition difficult.

The target statistics including their locations, speeds, accelerations etc., are obtained through a vision-based tracking algorithm as described in [55]. The individual trajectories are sampled at discrete intervals to obtain a string of primitive events or actions. An action is computed based on the local motion as well as the spatial location.

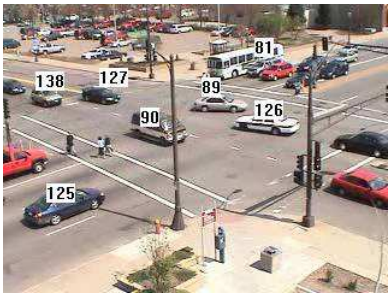
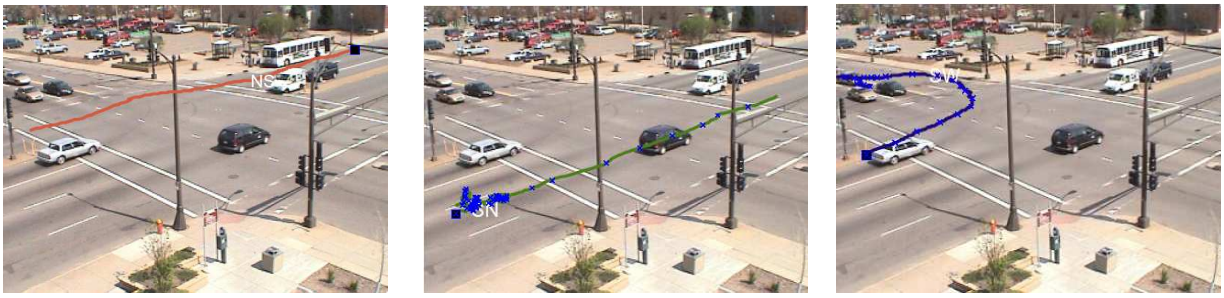


Figure 3.1: An example traffic scene used in the experiments.

3.2 Results

Fig. 3.2 and Fig. 3.3 show some examples of event classifications for different trajectories.[1]Detection of atypical events including U-turns and unusual motion paths are shown in Fig. 3.4. An advantage of applying the SCFGs is that a trajectory can be classified even with partial information as shown in Fig. 3.5, where only part of the vehicle's trajectory was available due to a large occlusion along the remaining trajectory of the vehicle.



(a) North-south motion.

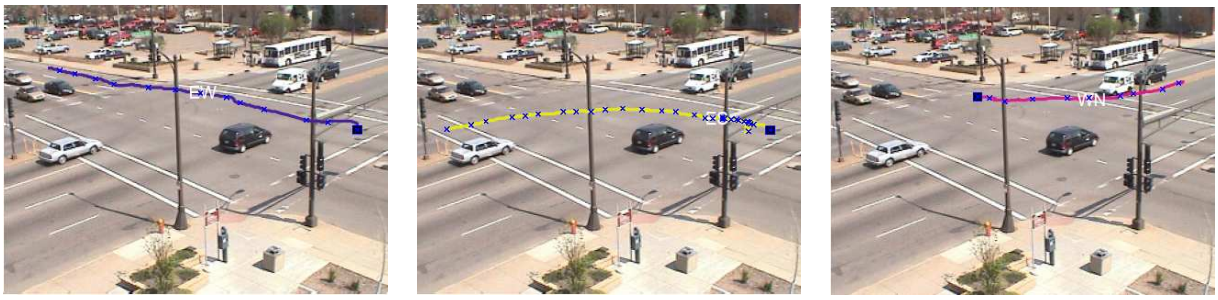
(b) South-north motion for a stop-n-go vehicle.

(c) South-west motion.

Figure 3.2: Trajectory classification by the SCFG for event classes north-south, south-north, and south-west.

Fig. 3.6 illustrates the effect of the ratio of labeled to unlabeled examples on the generalization performance of the classifier. The x-axis corresponds to the ratio of the number of labeled to unlabeled examples. The maximum number of labeled examples was 60 and unlabeled 290. As can be seen, the effect of increasing the number of labeled examples is

[1]Although the classification results are depicted on the same image, these trajectories arise from different vehicles under different traffic conditions.



(a) East-west motion. (b) East-south motion. (c) West-north motion.

Figure 3.3: Trajectory classification by the SCFG for east-west, east-south, and west-north, motions.

(a) Atypical motion I. (b) Atypical motion II.

Figure 3.4: Example atypical motion detection.

that the margin of generalization risk of using only labeled examples and combining labeled and unlabeled examples is reduced. This means that the effect of unlabeled examples diminishes as the number of labeled examples is increased. However, adding unlabeled examples still improves performance. The risk is computed by testing the classification performance of all the learned grammars on a test-set different from those used in the training examples.

As a benchmark experiment, we compared the classification performance of the proposed SCFG with a spectral clustering method as presented in [31]. In the tests, 1069 trajectories were used for testing. Being an unsupervised clustering method, all the data-sets were directly presented to the spectral clustering algorithm. For training the SCFG, a total of 250 examples from a different data-set was used. Out of the 250, 50 trajectories were labeled or 5-6 examples on an

(a) East-south motion. (b) North-east motion. (c) South-west motion.

Figure 3.5: Accurate detection can be obtained even only when a small portion of the trajectory is visible.

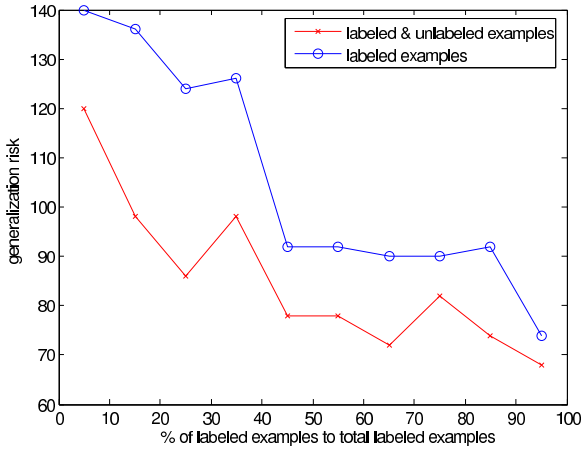


Figure 3.6: Generalization risk performance for training with varying numbers of labeled and unlabeled examples.

average per event class.[2] The results of the classification performance are depicted in Table. 3.1. One thing to note is that none of the test examples consisted of atypical motions such as U-turns, or reversing motions as these cannot be detected using the spectral clustering algorithm as it just makes use of the shape of the trajectories for clustering. Examples consisted of a mix of fully-visible and partially visible trajectories as were obtained from the tracking algorithm. Predictably, the clustering method fails when presented with partial trajectories. Besides, this algorithm will classify an unusual motion such as U-turn into one of the classes instead of detecting that as an outlier. With the SCFGs, it is possible to detect complex motions such as weaving motions which consists of multiple lane changes with very little additional knowledge and training. Detection of such complex motions is very challenging for both the clustering algo-

[2]One should note that not all the unsupervised examples are necessary for updating the grammar. The learning algorithm stops as soon as convergence results after the update from a few examples.

rithm as well as hidden Markov models. The latter will require complex formulations such as hierarchical models for such detection.

Classifier	*Correct*	*Incorrect*
SCFG	825 (77%)	244 (23%)
Spectral Clustering	669 (62%)	400 (38%)

Table 3.1: Results of classification on a data-set containing 1069 examples obtained from tracking video sequences in an outdoor scene.

Chapter 4

Discussion and Future Work

Overview

As depicted in the results, the proposed SCFG method is robust for obtaining event classifications in challenging environments such as traffic intersections. This chapter summarizes our findinds and proposes future work.

4.1 Findings and Future Work

The proposed methods can yield good classification performance even with partial information. The use of a skip or "lookahead" operation helps to deal with missing information such as resulting from temporary occlusions. A potential advantage of this event representation is that the method can easily scale to novel environments as it requires only a small amount of labeled data in addition to unlabeled data. Given the iterative nature of the learning algorithm, it can easily be converted into an any-time algorithm allowing the application of an on-line learning algorithm.

Classification accuracy is affected mostly by the ambiguities and missing information in the input data. For instance, it is possible that the grammar can learn incorrectly when presented with unlabeled data that contains noise. An interesting future extension would be to apply user feedback on the performance of classifier for grammar correction or to apply the method in conjunction with other classifiers to correct grammars using the results of other classifiers. Occasional non-

unique classification occurs in the case of partial trajectories when the trajectory lies in the profile of multiple classes.

Additionally, the spatial resolution of the cells used to discretize the image affects classification accuracy. Larger cells reduce the total number of spatial cells in the region, thereby increasing the overlap between the trajectories from various event classes resulting in increased ambiguities in classification. However, too small of a cell increases the number of total cells, thereby requiring a larger training data set for better generalization as well as increases the size of the event grammars. This in turn increases the time to parse and produce a match for a particular pattern. One possible direction of future work is to apply algorithms for automatically tuning the cell sizes to the observed data for better classification performance.

The current work examined the problem of event detection based on the activities of individual targets. One scope for future work is the problem of analyzing events arising from target interactions such as those arising in human activity monitoring, video annotation, and many more. This results in more complex and involves diverse events that need to be learned from the data.

Chapter 5

Conclusions

Overview

This chapter concludes this report by supplying an overview and our conclusions.

5.1 Concluding Remarks

This work introduced a novel technique for automatically learning the description of events from video sequences using stochastic context-free grammars. The events arising from the trajectory of the individual targets are represented as a string of actions. These strings are then parsed as stochastic context-free grammars which in turn are learned using a semi-supervised learning method. The learning method employs a simple, directed search-based approach to iteratively refine the grammars using conditional entropy as a heuristic. Experimental results obtained from real-world video sequences illustrate the classification performance of the proposed method.

References

[1] A.V. Aho, R. Sethi, and J. D. Ullman. *Compilers:* P*rinciples, techniques, and tools*. Addison-Wesley, 1986.

[2] A. Bobick and J. Davis. The recognition of human motion using temporal templates. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(3):257–267, March 2001.

[3] M. Brand and V. Kettnaker. Discovery and segmentation of activities in video. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(8):844–851, August 2000.

[4] M. Brand, N. Oliver, and A. Pentland. Coupled hidden Markov models for complex action recognition. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 994 – 999, June 1997.

[5] I. Cohen, F.G. Cozman, N. Sebe, M. C. Cirelo, and T.S. Huang. Semi-supervised learning of classifiers: theory, algorithms for Bayesian network classifiers and application to human-computer interaction. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(12):1553–1567, 2004.

[6] M. Collins. Parameter estimation for statistical parsing models: Theory and practice of distribution-free methods. In *IWPT*, 2001.

[7] F. Denis. Learning regular languages from simple positive examples. *Machine Learning*, 44(1/2):37–66, July 2001.

[8] D.M. Gavrilla and L.S. Davis. 3D model-based tracking of humans in action: a multi-view approach. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 73–80, 1996.

[9] R. Goldenberg, R. Kimmel, E.Rivlin, and M. Rudzsky. Behavior classification by eigen decomposition of periodic motions. *Pattern Recognition*, 38:1033–1043, 2005.

[10] S. Gong and T. Xiang. Recognition of group activities using dynamic probabilistic networks. In *Proc. IEEE Conf. Computer Vision*, volume 2, pages 742–749, 2003.

[11] Y. Gong, M. Han, W. Hua, and W. Xu. Maximum entropy model-based baseball highlight detection and classification. *Computer Vision and Image Understanding*, 96:181–199, 2004.

[12] Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In *Proc. Neural Information and Processing Systems*, volume 17, 2005.

[13] N. Haering, R.J. Qian, and M.I. Sezan. A semantic event-detection approach and its application to detecting hunts in wildlife video. *IEEE Trans. on Circuits and Systems for Video Technology*, 10(6):857–869, September 2000.

[14] A. Hakeem and M. Shah. Multiple agent event detection and representation in videos. In *AAAI*, pages 89–94, 2005.

[15] R. Hamid, A. Johnson, S. Batta, A. Bobick, C. Isbell, and G. Coleman. Detection and explanation of anamolous activities: representing activities as bags of event n-grams. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, volume 1, pages 1031–1038, June 2005.

[16] S. Hongeng and R. Nevatia. Large-scale event detection using semi-hidden Markov models. In *Proc. IEEE Conf. Computer Vision*, volume 2, 2003.

[17] M. Inoue and N. Ueda. Exploitation of unlabeled sequences in hidden Markov models. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 25(12):1570–1581, 2003.

[18] L. Itti and P. Baldi. A principled approach to detecting surprising events in video. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 631–637, 2005.

[19] Y.A. Ivanov and A.F. Bobick. Recognition of visual activities and interactions by stochastic parsing. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(8):852–872, 2000.

[20] F. Jelinek and J.D. Lafferty. Computation of the probability of initial substring generation by stochastic context-free grammars. *Computational Linguistics*, 17(3):315–323, 1991.

[21] N. Johnson and D. Hogg. Learning the distribution of object trajectories for event recognition. *Image and Vision Computing*, 14(8):609–615, August 1996.

[22] S.X. Ju, M.J. Black, and Y. Yacoob. Cardboard people: a parameterized model of articulated image motion. In *Proc. Intl. Conf. on Automatic Face and Gesture Recognition*, pages 16–22, October 1996.

[23] S. Kamijo, Y. Matsushita, K. Ikeuchi, and M. Sakauchi. Traffic monitoring and accident detection at intersections. *IEEE Trans. on Intelligent Transportation Systems*, 1(2):108–118, June 2000.

[24] Y. Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. In *Proc. IEEE Conf. Computer Vision*, volume 1, pages 166–173, Oct 2005.

[25] B. Keller and R. Lutz. Evolutionary induction of stochastic context free grammars. *Pattern Recognition*, 38:1393–1406, 2004.

[26] P. Kumar, S. Ranganath, W.M. Huang, and K. Sengupta. Framework for real-time behavior interpretation from traffic video. *IEEE Trans. on Intelligent Transportation Systems*, 6(1):43–53, March 2005.

[27] K. Lari and S.J. Young. The estimation of stochastic context-free grammars using the inside-outside algorithm. *Computer Speech and Language*, 4:35–56, 1990.

[28] B. Li, J.H. Errico, H. Pan, and I. Sezan. Bridging the semantic gap of sports video retrieval and summarization. *Journal of Visual Communication and Retrieval*, 15:393–424, August 2004.

[29] X. Li and F. M. Porikli. A hidden Markov model framework for traffic event detection using video features. In *Proc. IEEE Conf. on Image Processing*, volume 5, pages 2901–2904, October 2004.

[30] Y. Li, S. Narayanan, and C.C.J. Kuo. Content-based movie analysis and indexing based on audio visual cues. *IEEE Trans. on Circuits and Systems for Video Technology*, 14(8):1073–1085, August 2004.

[31] L.Z.Manor and M. Irani. Event-based analysis of video. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, volume 2, pages 123–130, 2001.

[32] G. Medioni, I. Cohen, F. Bremond, S. Hongeng, and R. Nevatia. Event detection and analysis from video streams. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(8):873–889, August 2001.

[33] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(7):696–710, July 1997.

[34] Darnell Moore and Irfan Essa. Recognizing multitasked activities from video using stochastic context-free grammar. In *Eighteenth national conference on Artificial Intelligence*, pages 770–776. American Association for Artificial Intelligence, 2002.

[35] M.R.Naphade and T.S.Huang. Discovering recurrent events in video using unsupervised methods. In *Proc. IEEE Conf. Image Processing*, volume 2, pages 13–16, 2002.

[36] M. R. Naphade, A. Garg, and T. S. Huang. Duration dependant input output hidden Markov models for audio-visual event detection. In *Proc. IEEE Intl. Conf. on Multimedia and Expo*, pages 253–256, Aug 2001.

[37] N.T. Nguyen, D.Q. Phung, S. Venkatesh, and H. Bui. Learning and detecting activities from movement trajectories using the hierarchical hidden Markov model. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, volume 2, pages 955–960, 2005.

[38] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using em. *Machine Learning*, 39:103–134, 2000.

[39] O.C.Jenkins and M. J. Matarić. Deriving action and behavior primitives from human motion data. In *Proc. IEEE Conf. Intelligent Robots and Systems*, pages 2551–2556, 2002.

[40] N. Oliver, A. Garg, and E. Horvitz. Layered representations for learning and inferring office activity from multiple sensory channels. *Computer Vision and Image Understanding*, 96(2):163–180, 2004.

[41] N. Oliver, B. Rosario, and A. Pentland. A Bayesian computer vision system for modeling human interactions. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(8):831–843, 2000.

[42] M. Osadchy and D. Keren. A rejection-based method for event detection in video. *IEEE Trans. on Circuits and Systems for Video Technology*, 14(4):534–541, April 2004.

[43] R. Parekh and V. Honavar. Automata inducation, grammar inference, and language acquisition. In Dale, Moisl, and Sommers, editors, *The Handbook of Natural Language Processing*, pages 727–764. Marcel Dekker Inc, New York, 2000.

[44] N. Peyrard and P. Bouthemy. Motion-based selection of relevant video segments for video summarization. *Multimedia Tools and Applications*, 26(3):259–276, August 2005.

[45] R. Pless. Image spaces and video trajectories: using Isomap to explore video sequences. In *Proc. IEEE Conf. Computer Vision*, volume 2, pages 1433–1440, 2003.

[46] R. Polana and R.C. Nelson. Detection and recognition of periodic, non-rigid motion. *Intl. Journal of Computer Vision*, 23(3):261–282, 1997.

[47] R.Mann, A.Jepson, and J. Siskind. The computational perception of scene dynamics. *Computer Vision and Image Understanding*, 65(2):113–128, February 1997.

[48] Y. Sakakibara. Efficient learning of context-free grammars from positive structural examples. *Information and Computation*, 97:23–60, 1992.

[49] Y. Sakakibara. Grammatical inference in bioinformatics. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(7):1051–1062, July 2005.

[50] J.M. Siskind. Grounding the lexical semantics of verbs in visual perception using force dynamics and event logic. *Journal of Artificial Intelligence Research*, 15:31–90, 2001.

[51] C. Stauffer. Automatic hierarchical classification using time-based co-occurrences. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, volume 2, pages 339–344, 1999.

[52] A. Stolcke. *Bayesian learning of probabilistic language models*. PhD thesis, University of California, Berkeley, 1994.

[53] A. Stolcke and S. Omohundro. Inducing probabilistic grammars by Bayesian model merging. In *Proc. Second Intl. Colloquium on Grammatical Inference*, pages 106–118, 1994.

[54] C.M. Taskiran, I. Pollak, C.A. Bouman, and E.J. Delp. Stochastic models of video structure for program genre detection. In *Visual Content Processing and Representation, Lecture Notes in Computer Science*, volume 2849, 2003.

[55] H. Veeraraghavan, P. Schrater, and N.P. Papanikolopoulos. Robust integration of motion, color, and geometry for target detection and tracking. *IEEE Trans. on Intelligent Transportation Systems*, 103(2):121–138, August 2006.

[56] T. Wada and T. Matsuyama. Multiobject behavior recognition by event driven selective attention method. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(8):873–887, August 2000.

[57] G. Xu, Y. Ma, H. Zhang, and S. Yang. A HMM based semantic analysis framework for sports game event detection. In *Proc. IEEE Conf. on Image Processing*, volume 1, pages 25–28, Sept 2003.

[58] Y. Yacoob and M. Black. Parameterized modeling and recognition of activities. *Computer Vision and Image Understanding*, 73(2):232–247, 1999.

[59] Y.Sakakibara. Learning context-free grammars using tabular representations. *Pattern Recognition*, 38:1372–1383, 2004.

[60] D. Zhong, D.G. Perez, S. Bengio, and I. McCowan. Semi-supervised adapted HMMs for unusual event detection. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2005.

[61] H. Zhong, J. Shi, and M. Visontai. Detecting unusual activities in video. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2004.