Learning Dynamic Event Descriptions in Image Sequences

Veeraraghavan, H., Schrater, P.,R., Papanikolopoulos, N.
CVPR 2007, Minneapolis MN

Abstract

Automatic detection of dynamic events in video sequences has a variety of applications including visual surveillance and monitoring, video highlight extraction, intelligent transportation systems, video summarization, and many more. The main challenge in most real-world applications is the learning of the event descriptors from limited data in the presence of noise and variations in the occurrence of such events. The main contribution of this work is a semi-supervised learning method in the aforementioned setting for detecting dynamic events in video sequences. Concretely we introduce a stochastic context-free grammar approach for representing the events and learn the event descriptors using an entropy minimization-based semi-supervised method. Experimental results demonstrating the efficacy of the learning algorithm and the event detection method applied to real-world video sequences are presented.

Keywords

Event detection, stochastic context-free grammars, semi-supervised learning, entropy minimization.

1 Introduction

Dynamic event detection from video sequences is a fundamental problem in computer vision with several applications including, video surveillance and monitoring, video indexing and highlight extraction, intelligent transportation systems, and many more. Spatio-temporal trajectories such as a vehicle or a robot moving with varying speeds give rise to dynamic events. For example, the dynamic event shown in Fig. 1 consists of a vehicle moving from south to north through an intersection. As illustrated in the figure, this event represents a *class* of trajectories, as vehicle trajectories can vary in several ways, including speeds, lane changes, stop-n-go motion without altering the basic event type. Real-world scenes present additional complexities in the form of ambiguous data (arising from noise) and limited user-labeled data. Automatic learning and the detection of dynamic events under these settings is the main contribution of this work.

The standard approach to dynamic event detection using state space models such as the hidden Markov models (HMM) are not applicable or atleast require complex formulations such as [5, 13]. Additionally, such models require prohibitively large state space for representing the events, which in turn requires a large data-set for training. On the other hand, Stochastic Context-Free Grammars(SCFG) [16, 6] with their flexible representation provide more expressibility and require simple models for the event representation. This in turn allows training using a small labeled data-set even in complex environments such as outdoor scenes as depicted by our results.

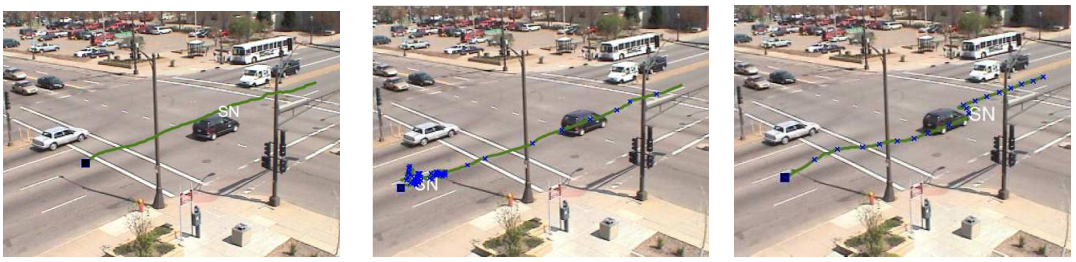
While SCFGs have been applied to a limited extent to event detection from image sequences [6, 11], little attention has been paid to learning the grammar from data. This works addresses this issue given a knowledge of the grammar structure. The advantage in automatically learning the event descriptors or the grammar from the data is that such an approach scales easily to novel scenes with minimal user provided knowledge.

This paper is organized as follows: After introducing the problem in Section 1, the learning problem is formally introduced in Section 2. A brief background of the learning method is presented in Section 3 followed by a survey of related works in Section 4. The event detection method using the SCFGs and the learning algorithm are described in Section 5. Section 6 presents some experimental results of event detection on some real-world example trajectories followed by their discussion in Section 7. Finally, Section 8 concludes the paper.

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 [9] is the optimization algorithm for estimating the rule probabilities. However, automatically learning all the parameters, namely, the terminal symbols, the non-terminals, the productions, and their probabilities from data is generally a much difficult problem as the structure of the grammar is unknown. However, when at-least part of the structure is available to the learner, it has been shown that grammars can be induced in polynomial time [14, 16]. In this work, the structure of the grammar is supposed to be available beforehand as bracketed pairs of actions. The main challenge however is learning from data contaminated by noise as is common in vision-based inference from uncontrolled image se-



(a) Typical south-north motion. (b) Stop-n-go south-north motion. (c) Lane change south-north motion.

Figure 1: Example categories of south-north motion. As shown, several sub-categories or events arise for the same south-north motion.

quences. Noisy inference results from ambiguities in interpreting occlusions, illumination changes, and other factors resulting from unmodeled environmental effects.

3 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.

3.1 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 [2] 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 it's 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,

$$H(y|S) = -\sum_{i=1}^{k} p(y|S=k, G_k) log(p(y|S=k; G_k)) \quad (1)$$

where y is the output label $[0-1]$, S is the input spatio-temporal pattern, G_k is the current grammar for class k.

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. Another measure applicable for the stopping condition is the empirical conditional classificiation entropy, expressed as,

$$H(y) = \sum_{i=1}^{M} H(y|S_i) \quad (2)$$

where there are M unlabeled patterns.

4 Related Work

Approaches to detecting spatio-temporal events range from dimensionality reduction methods such as [10, 12], to the frequently used state space models such as the hidden Markov models (HMM) and their variations. Examples of the HMM-based approaches applied to detecting activities in video sequences include [7, 1, 5]. The main limitation of the dimensionality methods stems from the use of unsupervised clustering to guide the dimensionality reduction which in turn fails to weight the data differently with respect to noise. Furthermore, these approaches also require a lot of data to obtain a reasonable estimate of the event profiles. Similarly, state space model approaches such as HMMs require non-typical formulations such as time-duration HMMs for modelling the varying temporal scales of the events. The complex models coupled with the generative model setting of HMMs increases the burden of training particularly in terms of a larger training data set.

Context-free grammar approaches have been recently applied for event recognition in video sequences. The flexibility of representation afforded by these methods allows one to model a larger set

of variations in the data for a particular kind of event. Hamid *et al.* [4] 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 [3] 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 [6, 11]. Ivanov and Bobick applied SCFGs [6] 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 HMM) 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 [11] using addition, deletion, and insertion operations similar to those used in edit distances for matching strings. The events were detected for a small set of fixed entities in the scene. Until now, to the author's knowledge all video-based event detection approaches employing variants of probabilistic grammars are restricted to recognition or classification of activities using a pre-specified grammar. This work addresses the problem of automatically learning the grammar from the image data using a semi-supervised learning approach. Most of the work on learning the grammars exists in language modeling [8, 15] and bioinformatics [18], albeit using fully labeled data.

5 Event Detection using Stochastic Context-Free Grammars

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. 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 example string is depicted in Fig. 3, where $C1, C5, C6$ correspond to the discrete spatial cells and (*straight, fast)*correspond to the local motion.

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

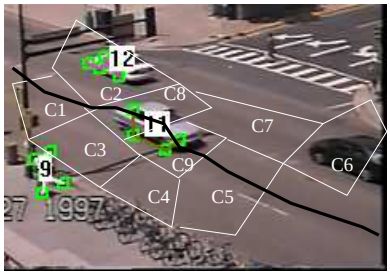


Figure 2: Example cell-based representation of the spatial region. The cells are shown as white regions with the corresponding labels. An example trajectory of a lane-changing vehicle is shown in black.

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

$$\Delta = \sum_{i=1}^{N} \prod_{j=1}^{k} p(y(i)|s(i); \Omega_j). \quad (3)$$

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. (3) becomes,

$$H(y|S) = \sum_{i=1}^{N} \sum_{j=1}^{k} p(y(i)|s(i); \Omega_j) \log \frac{1}{p(y(i)|s(i); \Omega_j)}. \quad (4)$$

where $S = s_1, \ldots, s_N$. Again, the empirical conditional entropy is the lowest when the co-dependance of y and s is high. Hence, the algorithm consists of refining the class grammars iteratively such that the empirical conditional entropy is minimized in each step and repeated until convergence. The basic learning algorithm is summarized in Table. 1.

As depicted in Table. 1, after learning a preliminary model with a few labeled training examples, the SCFG model for each class is iteratively refined until convergence. The computational complexity of the learning algorithm is polynomial, with the worst case complexity $O(n^2k + C)$ where n is the number of training examples, and k is the number of classes. Assuming that

```
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)
           newE = sum(E(i=1:M)) //recompute empirical conditional entropy
           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 1: Algorithm for grammar update. The grammar for the typical classes is updated incrementally using the strings with the least entropy.

$k << n$, the complexity of the algorithm is $O(n^2)$.

Once the grammar is revealed, namely, the set of terminals, non-terminals, and productions, the production probabilities can be estimated using the standard optimization techniques such as [9].

As mentioned earlier, the grammar structure is assumed to be available before-hand in the form of bracketed expressions. An example is depicted in Fig. 3. 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 [16].

The only prior knowledge the learning algorithm requires is some knowledge of the structure of the grammar 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.

5.3 Event Classification and Error Recovery

Once the grammar for each class is learned, a novel pattern is classified by parsing it with all the available grammars. The grammar which produces the successful parse of the pattern is attributed as producing the pattern. In the case of multiple successful parses, the grammar that required the least number of skips of the strings is attributed the classification. Otherwise, ties are broken arbitrarily.

The skip or the "lookahead" operation is mainly used for error recovery. For instance, some of the sub-strings in a given pattern may be erroneous owing to the target temporarily covered by an occlusion. In such a case it is not possible to generate a successful parse and such strings are skipped. However, a penalty is also associated with the skips to prevent all patterns being classified

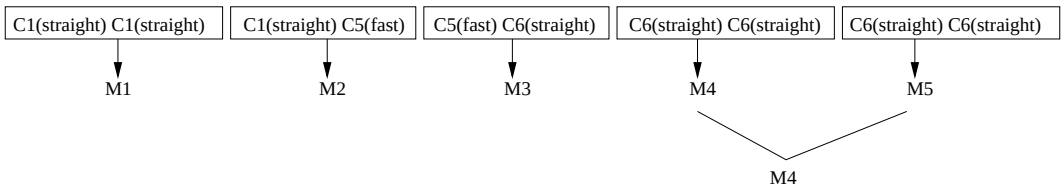


Figure 3: An example action sequence with bracketed structure and non-terminal creation. The boxes around the strings represent the bracketed structure. The terminals consist of the spatial cell occupied by the target such as C1, and the corresponding local motion, such as *straight, fast, slow, left, right turn* . The non-terminals are depicted as M1, M2, M3, etc.

into all classes. The penalty is computed as,

$$\epsilon_l = exp(-(\eta)/L) \tag{5}$$

where η is the number of lookahead steps and L is the total length of the string. The larger the number of skips the smaller the ϵ_l. This is similar to the error recovery scheme used in [11]. Unlike [11], currently we don't use the insert and delete operations and leave that as part of future work.

6 Experimental Results

Objective The objective of the experiments was to test the efficacy of the proposed learning method on real-world image sequences.

6.1 Experiment Description

For the experiments we chose scenes from outdoor traffic intersections such as depicted in the Fig. 4. 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 makes target localization a challenging task. The resulting ambiguity in the observed data in addition to the significant overlaps between various events makes event recognition difficult.

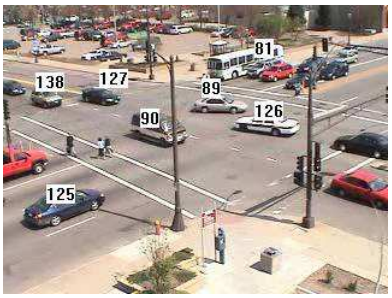


Figure 4: An example traffic scene used in the experiments.

The target statistics including their locations, speeds, accelerations etc., are obtained through a vision-based tracking algorithm as described in [17]. 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.

6.2 Results

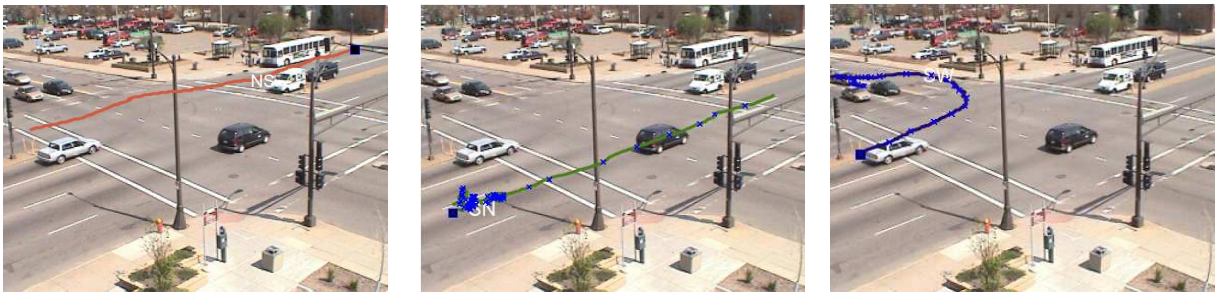
Fig. 5 and Fig. 6 show some examples of event classifications for different trajectories.[2]Detection of atypical events including U-turns and unusual motion paths are shown in Fig. 7. An advantage of applying the SCFGs is that a trajectory can be classified even with partial information as shown in Fig. 8, where only part of the vehicle's trajectory was available due to a large occlusion along the remaining trajectory of the vehicle.

Fig. 9 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 that the margin of generalization risk of using only labeled examples and combining labeled, 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 testset 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 [10]. In the tests, 1069 trajectories were used for testing. Being an unsupervised clustering method, all the datasets 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 average per event class.[3] The results of the classification performance are depicted in Table. 2. 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 was obtained from the tracking algorithm. Predictably, the clustering method fails when presented with partial

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

[3]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.



(a) North-south motion. (b) South-north motion for a stop-n-go vehicle. (c) South-west motion.

Figure 5: Trajectory classification by the SCFG for event classes north-south, south-north, and south-west.

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

Figure 6: Trajectory classification by the SCFG for east-west, east-south, and west-north motions.

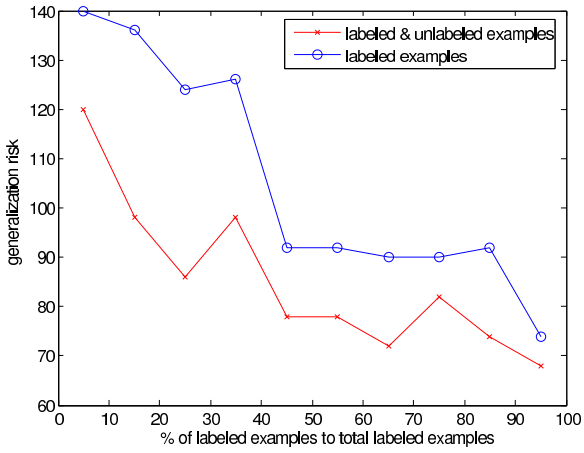


Figure 9: Generalization risk performance for training with varying numbers of labeled and unlabeled examples.

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

7 Discussion and Future Work

As depicted in the results, the proposed SCFG method is robust for obtaining event classifications in challenging environments such as traffic intersections. The method 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

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

Figure 7: Example atypical motion detection.

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

Figure 8: Accurate detection can be obtained even only when a small portion of the trajectory is visible.

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 diverse events that need to be learned from the data.

8 Conclusions

This work developed stochastic context-free grammars (SCFG) for event detection in challenging outdoor video sequences. The main contribution of this work is a semi-supervised learning algorithm applied to the SCFG for learning the event categorizations in a given scene. Experimental results on real-world image sequences show the robust performance of this method.

9 Acknowledgements

References

[1] 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.

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

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

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

Table 2: Results of classification on a data-set containing 1069 examples obtained from tracking video sequences in an outdoor scene.

[4] R. Hamid, A. Johnson, S. Batta, A. Bobick, C. Isbell, and G. Coleman. Detection and explanation of anomalous 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.

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

[6] 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.

[7] 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.

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

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] Y. Sakakibara. Learning context-free grammars from structural data in polynomial time. *Theoretical Computer Science*, 76:223–242, 1990.

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

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

[17] xxx. xx. *x*.

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