Learning Dynamic Event Descriptions in Image Sequences

Harini Veeraraghavan Nikolaos Papanikolopoulos* Paul Schrater
Department of Computer Science and Engineering
University of Minnesota
{harini,schrater,npapas}@cs.umn.edu

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. Learning an accurate description of the various events in real-world scenes is challenging owing to the limited user-labeled data as well as the large variations in the pattern of the events. Pattern differences arise either due to the nature of the events themselves such as the spatio-temporal events or due to missing or ambiguous data interpretation using computer vision methods. In this work, we introduce a novel method for representing and classifying events in video sequences using reversible context-free grammars. The grammars are learned using a semi-supervised learning method. More concretely, by using the classification entropy as a heuristic cost function, the grammars are iteratively learned using a search method. Experimental results demonstrating the efficacy of the learning algorithm and the event detection method applied to traffic video sequences are presented.

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 are a class of events that are commonly observed in video sequences. The said events arise due to varying spatial occupancies and temporal scales of moving targets in a scene. Additionally, real-world scenes present challenges to a learning system in the form of inaccurate data interpretation from uncontrolled scenes as well as limited user labeled data. Automatically learning the event descriptions under the above mentioned settings is the main contribution of this work.

The standard approach to dynamic event detection consists of employing state space models such as the hidden Markov models (HMM). While robust, learning such models for complex data requires hierarchical [4, 12], as well as variable duration models. This in turn increases the number of design parameters for setting up the model as well as the free parameters for model estimation. On the other hand, context-free grammars (CFGs) with their flexible representation provide more expressive power with more straightforward design.

While SCFGs have been applied to a limited extent for event detection from image sequences [5, 9], little attention has been paid to learning the grammar from data. This work addresses the problem of learning the event grammars, namely, the set of terminal symbols, the non-terminals, the rules, and the rule probabilities from the data.

This paper is organized as follows: After introducing the problem in Section 1, related works are presented in Section 2. The learning problem is then formally introduced in Section 3. The semi-supervised learning method is then discussed in Section 4. Section 5 presents some experimental results of event detection on a real-world application, namely, traffic monitoring from video sequences. Finally, Section 7 concludes the paper.

2. Related Work

Approaches to detecting spatio-temporal events range from dimensionality reduction methods such as in [8, 10], to the frequently used state space models such as the hidden Markov models (HMM) and their variations [4, 11]. While robust, state space models such as HMMs require hierarchical and time-duration modelling for representing events with varying temporal and spatial scales. This in turn increases the complexity of the design.

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. Ivanov and Bobick [5] employed stochastic context-free grammars (SCFG) for recognizing gestures by combining a HMM at lower level with SCFG. Recently, Ryoo and Aggarwal [13] combined Bayesian networks and HMM in the lower levels with events represented as context-free grammars for recognizing complex human interactions. Hamid *et al*. [2] developed an approach to detect anomalous activites in video sequences

1-4244-1180-7/07/$25.00 ©2007 IEEE

using hand-coded tri-grams. Similarly, Hakeem and Shah [1] employed graphical networks with hand-coded grammars for detecting activites in videos. In order to deal with ambiguities in image-based inference, Moore and Essa [9] introduced an interesting approach using addition, deletion, and insertion operations in the SCFGs.

Until now, to the authors' 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 both structure, namely, the set of rules, non-terminals, and terminal symbols, and the parameters, namely, the rule probabilities. Most previous work in learning grammars, both structure and parameters exists in language modelling [6, 16], and bioinformatics [15]. An interesting approach to estimating the parameters of the grammar for learning good estimates of the histogram probabilities is in [3].

3. 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 [7] is the optimization algorithm for estimating the rule probabilities. However, automatically learning the structure, namely, the terminal symbols, the non-terminals, the productions, and the parameters of the grammar is a much difficult problem. However, when certain assumptions can be made about the data such as the availability of bracketting information or partially available grammar structure, the grammars can be induced in polynomial time [14, 17].

In this work, we assume that the data contains bracketting information and that the grammars are structurally reversible. A structurally reversible grammar is one of type where among all the non-terminals that might derive a given terminal string, no one is an extension of the other. In other words representing upper case English alphabets as non-terminals and Greek letters as terminals, when, $A \rightarrow \alpha$ and $B \rightarrow \beta$, then $A = B$. Again, when $A \rightarrow \alpha B \beta$ and $B \rightarrow \alpha C \beta$, then $B = C$.

4. Event Detection Using Stochastic Context-Free Grammars

4.1. 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 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. 1. The cells can either be laid out by the user or randomly generated.

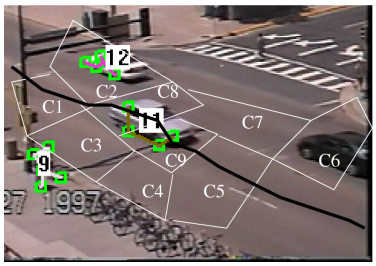


Figure 1. Example cell-based representation of the spatial region. The cells are shown as white regions with the corresponding labels. An example trajectory of a lane-changing vehicle is shown in black.

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, where $C1, C5, C6$ correspond to the discrete spatial cells and (*straight, fast)*correspond to the local motion. An action is $C1(straight)$ for example, while a bracketted set of actions are represented in boxes as $C1(straight)C5(fast)$ in the Fig. 2. The entire set of actions represents a string pattern.

4.2. 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 pattern $s = a_1, a_2, \ldots, a_n$, where $a_1, a_2, \ldots, a_n$ are the set of actions in the pattern, 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,

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

Entropy is the clearest way of characterizing the uncertainty in the posterior probabilities of the classification labels y for an input pattern $s = a_1, \ldots, a_M$. $a_1, \ldots, a_M$ are the sampled actions (spatial location and motion) obtained from the trajectory. In other words, an entropy zero corresponds to perfect classification, or the case when utmost one class is attributable

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

to the given input string. The conditional classification entropy represented as shown in Eqn. 2, provides a convenient measure for representing the cost for augmenting the grammar of a class using a particular example.

$$H(y|s=\{a_1,\ldots,a_M\}) = -\sum_{i=1}^{k}\sum_{j=1}^{M} p(y|a_j,G_k)log(p(y|a_j;G_k)) \quad (2)$$

s is the string pattern consisting of component sampled actions $s = a_1,\ldots,a_M$, while y is the output label for the pattern s. There are k classes of events and G_k is the grammar for class k.

The learning method iteratively searches through the set of unlabeled examples, and adds the example with the least conditional classification entropy. The conditional classification entropy is essentially a heuristic cost metric for guiding the search algorithm, where, the algorithm is essentially a best-first search. As a result, the solution produced by the search algorithm is optimal when the conditional entropy is an admissible heuristic. However, it is not possible to guarantee that the chosen cost metric will always overestimate the cost to goal. Hence, the solution is not guaranteed to be optimal. However, in this work, our focus is on obtaining a satisfysing solution if not an optimal one.

Another useful metric for the search algorithm is the empirical conditional entropy. Assuming uniform prior to all classes, $1,\ldots,k$, the said entropy can be represented as,

$$H(y|S) = \sum_{i=1}^{N}\sum_{j=1}^{k}\sum_{l=1}^{M_i} p(y(i)|a_l^i;G_j)\log\frac{1}{p(y(i)|a_l^i;G_j)}. \quad (3)$$

where $S = s_1,\ldots,s_N$ is the set of patterns. a_l^i refers to an action a_l occurring in the pattern s_i. Again, the empirical conditional entropy is the lowest when the co-dependence of the label y and the patterns S are high. The algorithm makes use of the empirical conditional entropy as a stopping criterion for the search algorithm. Thus, the algorithm halts refinement when either no more examples below a prespecified conditional classification entropy threshold are left for update, or when the empirical conditional classification entropy of the remaining examples falls below a set threshold E_{th}. The basic 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. Using the set of examples associated to a class, the grammar structure is built from the terminals. Terminals may be merged to form non-terminals, following which, the rule probabilities are computed. This is done by computing histogram counts of rules applied for parsing.

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. 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 [17].

Input: Unlabeled Examples $U = 1,\ldots,N$,
Labeled Examples $L = 1,\ldots,N_l$,
Classes k
Output: Grammar Set $G_1, G_2, \ldots, G_k$

GRAMMAR-LEARN(U, L)
$G \leftarrow$ UPDATE-GRAMMAR$(L, 1, \ldots, k)$
compute empirical conditional entropy E_h
$E_h \leftarrow \sum_{i=1}^{N}\sum_{j=1}^{k}\sum_{l=1}^{M_i} p(y(i)|a_l^i;G_j)\times$
$\log\frac{1}{p(y(i)|a_l^i;G_j)}$
while $U \neq \{\} \vee E_h > E_{th}$
do for $i \leftarrow 1 \rightarrow N$
  do $classU(i) \leftarrow$ CLASSIFY(U_i, G)
    compute conditional classification entropy h_i
    $h_i \leftarrow \sum_{i=1}^{k}\sum_{j=1}^{M} p(y|a_j,G_k)log(p(y|a_j;G_k))$
  $CHOOSE\ U_i\ S.T.\ h_i == MIN(h_{1,\ldots,i})$
  $G' \leftarrow$ UPDATE-GRAMMAR$(U, classU(i))$
  $E_{hnew} \leftarrow \sum_{i=1}^{N}\sum_{j=1}^{k}\sum_{l=1}^{M_i} p(y(i)|a_l^i;G'_j)\times$
  $\log\frac{1}{p(y(i)|a_l^i;G'_j)}$
  if $E_{hnew} - E_h > \alpha$
    then *reset Grammar to* G
      $h_i \leftarrow \infty$
      $GOTO\ 10$
    else $E_h \leftarrow E_{hnew}$
      $GOTO\ 5$

Table 1. Algorithm for grammar update. The grammar for the typical classes is updated incrementally using the strings with the least entropy.

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.

4.3. 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. For parsing we make use of the standard Earley parsing method. In the case of multiple classifications, ties are broken arbitrarily. In order to allow for missing data, the algorithm skips sub-strings for obtaining

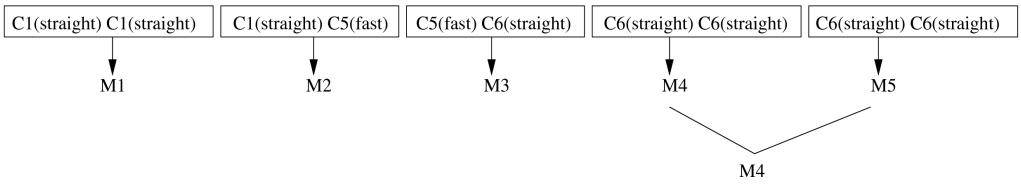


Figure 2. An example action sequence with bracketed structure and non-terminal creation. The boxes around the strings represent the bracketed structure. The terminals consist of the spatial cell occupied by the target such as C1, and the corresponding local motion, such as *straight, fast, slow, left, right turn* . The non-terminals are depicted as M1, M2, M3, etc.

a successful parse. The adopted method is similar to the error recovery methods used in [9] with the exception that the algorithm just implements the skip mechanism instead of all the skip, add, and delete methods.

5. Experimental Results

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

5.1. Experiment Description

For the experiments we chose scenes from outdoor traffic intersections such as depicted in the Fig. 3. 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

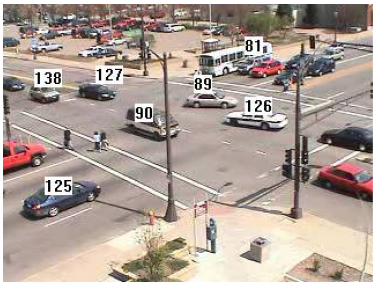


Figure 3. An example traffic scene used in the experiments.

target statistics including their locations, speeds, accelerations *etc*. are obtained through a vision-based tracking algorithm as described in [18]. 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.

5.2. Results

Fig. 4 and Fig. 5 show some examples of event classifications for different trajectories.[2] An advantage of applying the SCFGs is that a trajectory can be classified even with partial information as shown in Fig. 6, where only part of the vehicle's trajectory was available due to a large occlusion along the remaining trajectory of the vehicle.

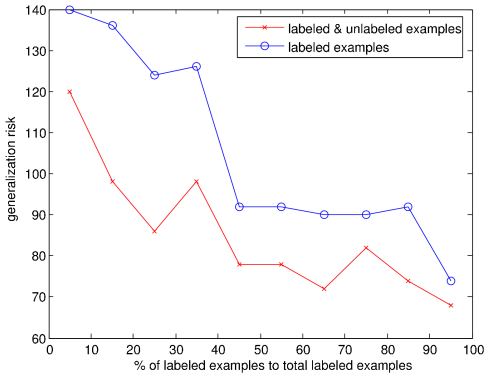


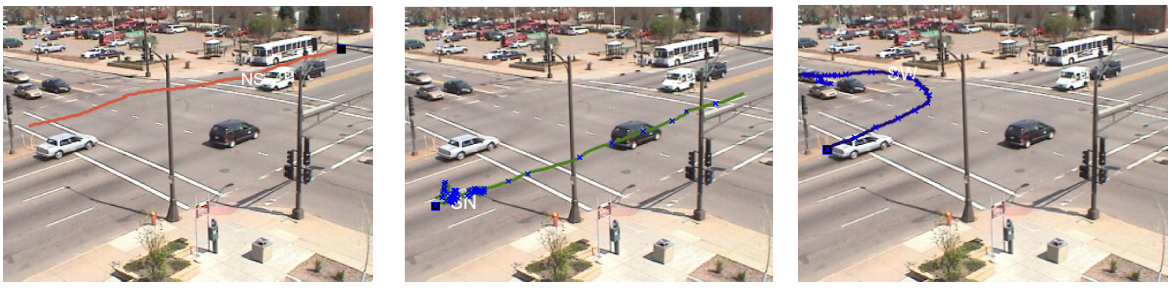
Figure 7. Generalization risk performance for training with varying numbers of labeled and unlabeled examples.

Fig. 7 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 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 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 [8]. 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]

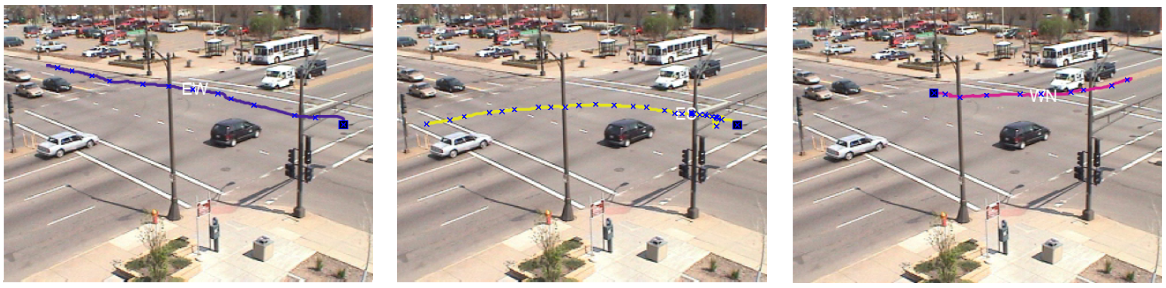
[2] Although the classification results are depicted on the same image, these trajectories arise from different vehicles under different traffic conditions.

[3] One should note that not all the unsupervised examples are necessary for updating the grammar. The learning algorithm stops as soon as convergence



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

Figure 4. 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 5. Trajectory classification by the SCFG for east-west, east-south, and west-north motions.

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. While one can compare the performance of the algorithm against anomalous activity recognition methods such as [19], we leave that as work for future. 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. With the SCFGs, it is possible to detect complex motions including weaving motions such as those resulting from lane changes. hierarchical models for such detection.

6. Discussion and Future Work

The SCFG-based event detection method can provide robust classification in challenging environments even when presented with partial information. As shown in the results, it is possible to extract reliable grammars for event recognition with a small set of labeled examples and a relatively small set of unlabeled examples. An advantage of this is that the method can easily scale to novel environments. The conditional classification entropy serves as a reasonable cost criterion to search for new grammars. However, since it is not possible to guarantee the feasibility of the heuristic, the algorithm is not guaranteed to generate the optimal grammar. However, in this work, our focus is obtaining a reasonable grammar for the data rather than on obtaining the optimal one.

The comparison of the performance of the results with the spectral clustering algorithm was merely done to assess the performance of the algorithm in comparison to another activity recognition algorithm. A better alternative for comparisons would involve other structure learning algorithms such as hidden Markov models. This work considered the feasibility of the learning approach for event detection. Extensive experiments on more real-world scenes and other applications including human activity recognition would be necessary to verify the applicability of the event detection method.

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.

7. Conclusions

This work presented a stochastic context-free grammar (SCFG) based method for event detection and classification in real-world image sequences. The main contribution of this work is a search-based iterative learning algorithm for learning the grammar structure and parameters for each class of motion using a semi-supervised learning strategy. Results demonstrating the feasibility of the learning algorithm and its applicability in real-world image sequences are presented.

results after the update from a few examples.

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

Figure 6. Accurate detection can be obtained even only when a small portion of the trajectory is visible.

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.

8. Acknowledgements

The authors would like to thank the anonymous reviewers for their comments. This work has been supported in part by the National Science Foundation through grant IIS-0219863, Architecture Technology Corporation, the Minnesota Department of Transportation, and the ITS Institute at the University of Minnesota.

References

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

[2] 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.

[3] B. Heeringa and T. Oates. Two algorithms for learning the parameters of stochastic context-free grammars. In *Working Notes of the 2001 AAAI Fall Symposium on Using Uncertainty Within Computation*, pages 58–62, 2001.

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

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

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

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

[8] 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.

[9] D. Moore and I. 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.

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

[11] N. Nguyen, D. 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.

[12] 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.

[13] M. S. Ryoo and J. K. Aggarwal. Recognition of composite human activities through context-free grammar based representation. In *Proc. of Computer Vision and Pattern Recognition*, volume 2, pages 1708–1718, 2006.

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

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

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

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

[18] H. Veeraraghavan, P. Schrater, and N. 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.

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