Temporal Structure Models for Event Recognition

John Greenall
jhs1jpg@leeds.ac.uk
Anthony G. Cohn
A.G.Cohn@leeds.ac.uk
David C. Hogg
D.C.Hogg@leeds.ac.uk

School of Computing
University of Leeds
Leeds, UK

Abstract

In many areas of visual surveillance, the observed activity follows re-occurring patterns. This paper demonstrates how such patterns can be exploited to improve the detection rate of independent event detectors. We present a temporal model based on pairwise correlations between event timings, which efficiently exploits limited training data. This is combined with the response from potentially heterogeneous independent event detectors to improve the robustness of detections over extended sequences. We demonstrate the efficacy of our system with rigorous testing on a large real-world dataset of aircraft servicing operations. We describe the implementation of a binary classifier based on local histograms of optical flow which is used as the independent event detector in our experiments.

1 Introduction

Throughout this paper we talk about about recognising events within a *scenario*. In many event recognition domains this notion is quite natural. Television programs are divided into episodes, sports footage is separated into games. Even in domains where video input is continuous such as traffic monitoring, there may be natural ways to divide the footage up i.e. morning commuter rush, midday lull, evening rush etc. Our method is applicable in any domain where such a notion of a scenario is natural, and there exists some repeated temporal structure. This work introduces a novel method for event recognition in structured domains. We define a model for efficiently combining the response from independent event detectors with a prior over the structure of inter-event timings. The model is designed to cope in situations where events of different types can potentially overlap and are not strictly ordered. These are precisely the conditions which cause problems for state-based models such as HMMs that attempt to encode the state of the world at each time step with a latent variable. In our optimization, the temporal midpoints and durations of event instances in a previously unseen sequence are the latent variables to be optimized. Our method, described in Section 3, only requires that the timings of at least some of the observed events are loosely correlated. The greater the degree of correlation, the greater the benefit will be; whilst the absence of any free parameters in our model makes it generic and easily applicable to other

BMVC 2011 http://dx.doi.org/10.5244/C.25.62

domains. We evaluate our method on a large real world dataset, implementing a state of the art activity classification system based on local spatiotemporal features as independent event detectors.

2 Related Work

Previous works on human action recognition can be divided into two broad categories: low-level, which focuses on the recognition of short basic actions normally involving a single participant and high-level, which focuses on recognising compound events involving several actions that are temporally correlated and normally involve multiple participants.

The state of the art methods in the low-level area tend to depend on local spatio-temporal features [4, 11]. The most popular datasets in this area have so far consisted of many short clips of actions from different classes: thus mapping the task to a classification problem. There have been some extensions and variations on these kind of methods which go slightly in trying to localize events in time and space [2, 5, 13]. However, there are few examples in the literature of these architectures being applied to the detection of sparse events in longer sequences.

Many existing approaches to high-level event recognition break complex activities down into a sequence of simpler atomic activities then exploit the temporal structure between these activities. This temporal model could take the form of a grammar [8], Bayesian Network [7] or a logical model [19] which effectively acts as a filter over fixed length detections originating from a lower-level detector. A disadvantage of this approach is that thresholding is generally necessary to decide when a detector should fire, meaning either many noisy detections must be permitted (which can lead to the interpretation space exploding) or else risk losing genuine detections. Another popular approach is to have latent variable(s) storing the state of the world at each timestep, and to model the probability distribution over transitions between *states*. HMMs are often deployed in this context due to their computational convenience. One of the main limitations of HMMs however is that the number of parameters in the model is proportional to the square of the number of hidden states. In scenarios where there are multiple overlapping processes, the state space of the hidden variable grows exponentially with the number of processes. This in turn increases the required amount of training data exponentially, which is a major problem in many domains. Extensions have been studied which mitigate the problem in certain special cases [15], but cannot be applied in general to cases with large numbers of overlapping events. Common to most recent works in high-level event recognition since [18], is the dependence on motion trackers as low-level input. The chief questions when considering the suitability of a trajectory-based approach to event detection for a particular domain are whether sufficiently reliable tracking is feasible and whether sufficient information about the events of interest is encoded in the trajectories alone [21]. For many domains, including the specific case of detecting servicing events on an airport apron, neither of these criteria can be said to be met by the current state of the art.

Methods which fall between the two broad categories outlined above include [14], which attempts to combine temporal structure and local features to recognise more complicated events. Here, a single event model consists of bag of word classifiers at different temporal scales, with some additional temporal context for each classifier. An alternative approach is presented in [17], where pairwise temporal relationships are appended to local features prior to quantization with the goal of retaining some of the temporal information (intra-event) that would otherwise be lost in local feature representation. Both of these methods focus

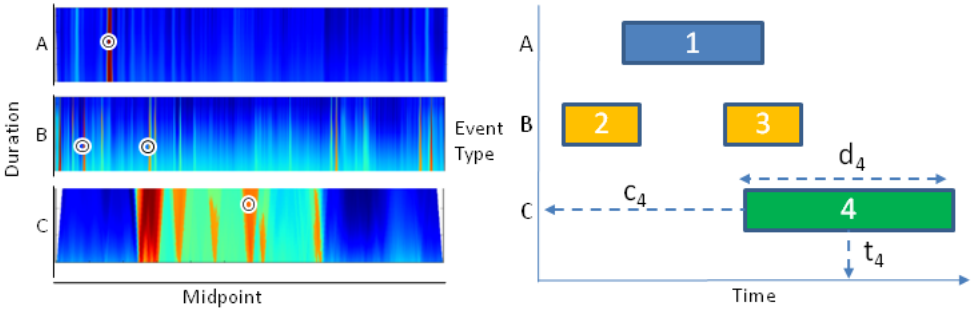

Figure 1: Left: Example response from 3 independent probabilistic event detectors. The detectors give a response for each possible interval in a sequence, with colour indicating the probability (red being high). We show four event instances localized on these likelihood streams. Right: Illustration of corresponding sequence scenario. The scenario is fully specified by event class vector $C = (c_1, ..., c_4)$ and temporal information $Y = ((t_1, d_1), ..., (t_4, d_4))$.

on the classification of relatively short actions in isolation. Our approach is complementary to these methods since either of them could be employed as independent event detectors in our system. Our model exploits temporal structure at the inter-event level to make a robust scenario recognition system.

3 Method

3.1 Constructing a Model for Scenario Recognition

Given a previously unseen video sequence of length T, we wish to determine the temporal midpoint, $t \in \mathscr{T} = \{1, .., T\}$, duration, $d \in \mathscr{T}$, and event class, $c \in \{1, ..., M\}$, of an unknown number, N, of event *instances*, from a fixed vocabulary of M event *classes*. This is depicted in Figure 1 (right). The scenario recognition problem over a sequence is challenging since it entails determining the correct number of event instances, N, then optimizating over these $3N$ random variables. We mitigate the problem of having an unknown N by introducing a'Null' state for each (t,d) so that $(t,d) \in \mathscr{T}^2 \cup Null$ as a mechanism for 'switching off' detections without having to dispose of the corresponding random variables entirely. To simplify matters, for the moment we assume there is at most one event of each type in each sequence. We address the problem of multiple instances of some events in Section 3.4. We are now able to fix the number of event instances in our scenario optimization to be M; fixing $C = (c_1, c_2, ..., c_M) = (1, 2, ..., M)$ and leaving $Y = ((t_1, d_1), ..., (t_M, d_M))$ to be optimised conditioned on C and the video sequence, X.

We assume that we have already trained a probabilistic event detector for each event class, and that these detectors will be run in parallel to evaluate every possible midpoint and duration over the discrete temporal domain.

Adopting a Bayesian perspective, the output of the independent event detectors can be treated as the likelihood of the relevant chunk of the observed video sequence, X, arising as a result of the given event scenario, which we term $p(X|C,Y,\theta)$, where θ is the set of parameters that define our model. Therefore $p(X|c_i, (t_i, d_i), \theta)$ is known for all possible

$(t_i, d_i) \in \mathscr{T}^2$. We set the probability of the event being *Null* as proportional to the complement of the maximum of the probability of it having occurred at any particular timestep. This enforces a suitable penalty for writing off the output of an event detector as noise and allows us to insert into our model a parameter, α_{c_i}, for each event class in order to vary the sensitivity of the detector. In our experiments, we found setting $\alpha_{c_i} = 1$ produced results near to the Equal Error Rate in most cases.

$$p(X|(t_i,d_i) = Null, c_i) = \alpha_{c_i}(1 - \max_{t_i,d_i}(p(X|(t_i,d_i) \in \mathscr{T}^2, c_i))) \tag{1}$$

We describe our implementation of a simple but powerful detector based on local spatio-temporal features in Section 4.2.

We can then concentrate on optimizing the posterior

$$p(Y|X,C,\theta) \propto p(X|C,Y,\theta)p(Y|C,\theta) \tag{2}$$

The distribution $p(Y|C,\theta)$ denotes the prior probability of a scenario given its event set and the model. By analogy to Pictoral Structure Models [6] from the object detection literature, we define our prior to be a tree-structured Markov Random Field (MRF). The general form for the joint distribution of such a prior can be written

$$p(Y|C,\theta) = \frac{\prod_{(i,j)\in E} p((t_i,d_i),(t_j,d_j)|c_i,c_j,\theta)}{\prod_{i\in V} p((t_i,d_i)|c_i,\theta)^{deg_i-1}} \tag{3}$$

where E is the edge set and V the vertices of the MRF. We do not want to give any preference over the absolute timing of any event in isolation, so therefore we set $p((t_i,d_i)|c_i,\theta)$ to be uniform, so that Equation 3 simplifies to.

$$p(Y|C,\theta) \propto \prod_{(i,j)\in E} p((t_i,d_i),(t_j,d_j)|c_i,c_j,\theta) \tag{4}$$

This prior has the following desirable characteristics. Firstly the tree structure ensures that the configuration is to some extent globally consistent since all random variables are (indirectly) connected. Secondly, the distribution is factorized into a product of pairwise terms which should reduce the demands on the amount of training data required to train the model. This is based on the assumption that the training data consists of a number of sequences, each of which contains a subset of the events in our vocabulary. Therefore the higher-order the relationship, the fewer the instances of co-occurrence in the dataset. Finally, the tree-structure allows exact inference on the prior, meaning we can quickly obtain globally optimal solutions.

3.2 Learning Inter-Event Priors

The first component of our prior which must be learnt is the set of $\frac{M(M+1)}{2}$ pairwise probability distributions $p((t_i,d_i),(t_j,d_j)|c_i,c_j,\theta)$ which we define to be probability distributions on the time differences between midpoints of instances of each possible pair of event classes (including the cases where $c_i = c_j$) appearing in the same sequence. Note that we omit any dependencies on the durations here for simplicity. In domains where the duration of events is small relative to the time between them, the information loss through this assumption will be negligible. Our initial experiments involved fitting a single Gaussian to the training data; but

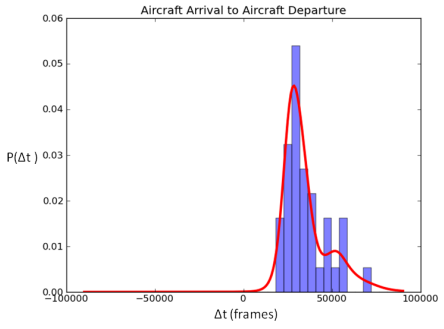

Figure 2: Left: Example of a learnt distribution over the time difference between two event classes in the airport domain. The training data is displayed in the form of a histogram. The distribution on the left is bimodal and skewed, but seems to be well approximated by the learnt distribution plotted in red. This a Gaussian Mixture Model of many overlapping components. Right: Depiction of the message passing scenario between two temporal midpoint random variables. We define the usefulness of the link between i and j to be inversely proportional to the lower of the two entropies $H[m_{j3}]$,$H[m_{i3}]$

having studied the data in our test domain, we observed that the single Gaussian is not a good approximator for many of the distributions. In some cases, we observe a distribution that is very skewed, perhaps indicating that one event is supposed to occur soon after another; it can never happen before, yet sometimes it does happen much later. In other cases, there may be two or more distinct modes. A Gaussian Mixture Model (GMM) with a sufficient number of components could model both of these effects. We follow the method of [16], which gives a full Bayesian treatment to the training of GMMs. This approach places a Dirichlet Process prior over the mixture weights of the model, as well as 'non-informative' priors over the mean and variance parameters. This way, the number of components is simultaneously optimized with the other parameters in the model. The Dirichlet Process priors make direct optimization intractable, so instead the posterior distribution over the parameter space is approximated through Gibbs sampling. We then use model averaging over 20 samples drawn over 1000 iterations (allowing 50 iterations for burn-in). The result of this procedure is a GMM with a large number of components which are highly-overlapped, but the overhead of evaluating these PDFs can be eliminated at execution time by the use of look-up tables. Figure 2 (left) gives an example of the kind of distributions learnt by this procedure.

Note that in the previous section we introduced *Null* states for the (t,d) variables. We now define the joint probability of a 'switched off' instance with any other event to be uniform:

$$p((t_i,d_i),(t_j,d_j)|c_i,c_j,\theta) = \frac{1}{T};\ (t_i,d_i) = Null \text{ or } (t_j,d_j) = Null \tag{5}$$

3.3 Dynamic Structure Assignment

The second component of the temporal prior which must be learnt is the edge set, E, of the MRF. We follow a procedure similar to that used in learning for Pictoral Structure Models [6], with the important difference being that the structure of our prior is dynamic (i.e. decided at recognition time), permitting us to formulate a data-dependent measure with which to rank

potential edges in the MRF. As we require our prior to be a tree-structured MRF, and the variables within Y are discrete, the MAP solution to Equation 2 may be obtained directly through the max-sum Belief Propagation (BP) algorithm [1]. Simplifying the prior to a tree structure will result in the loss of some information, so we need to ensure the edges we retain are the most informative. BP is a message passing algorithm and in this context, we believe it is reasonable to assume the most useful messages will be those of lowest entropy. Therefore to determine the structure of the tree, we evaluate the informativeness of each pairwise connection with the following equation.

$$I(i,j) = H[p(X|(t_i,d_i),c_i,\theta)\sum_{t_j} p(t_i,t_j|c_i,c_j,\theta)] \tag{6}$$

which is equivalent to the entropy of the initial message that would be passed from t_i to t_j (shown in figure 2 (right)). Note that this measure factors in the entropy of the detectors' responses over the sequence, meaning event instances with noisy observation likelihoods over a particular sequence are less likely to be connected to one another. Calculating I for each pair of event instances gives a $N \times N$ matrix, which we make symmetric by setting $I(i,j) \rightarrow \max(I(i,j),I(j,i))$. To this we apply Kruskal's algorithm [9] to construct the desired minimum spanning tree.

3.4 Scenario Recognition with Multiple Instances

In Section 3.1 we introduced the idea of a *Null* state to allow us to efficiently deal with the prior uncertainty over whether a particular event occurs within a sequence. Here we discuss our method for coping with the scenario recognition problem in its full generality: where multiple instances of the same event class may occur.

One obvious solution might be to set a hard upper limit on the number of instances of each particular class which could occur within a scenario, then to initialize the appropriate random variables before simply finding the MAP solution as in the single instance case. There is a problem with this approach since we take the observation likelihood for repeat instances of the same event class from the same detector. For this reason, we require instances of the same event class to be non-overlapping so that multiple instances don't point to the same activity. However, this is an impossible constraint to impose with a tree structured prior.

To solve the problem, we use a greedy procedure which is fast and proves to be effective. First we perform our exact optimization of Equation 2 allowing exactly one instance of each event class. For events which had instances localized successfully ($(t_i,d_i) \neq Null$), we create an additional instance for the next stage in our optimization. To prevent overlap, we fix the previously localized events by transforming their observation likelihood function to a Dirac delta function and set the observation likelihood of the new instance to zero at all (t_i,d_i) which would overlap the previously localized instance. We then restructure our tree-shaped prior and optimize again; repeating the process until we have a *Null* instance for each event class in our optimization.

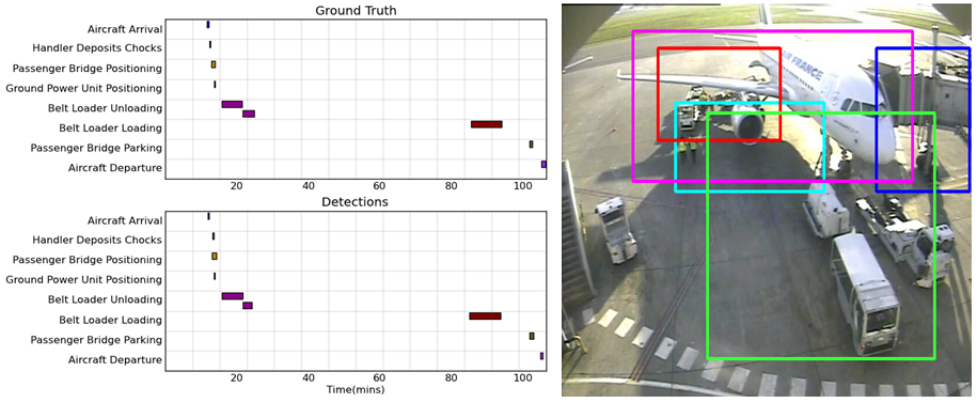

Figure 3: Left: Gantt chart plot showing successful detection results from one of our 37 sequences. In this case, the detected solution almost perfectly matches the ground truth. Right: Snapshot taken midway through a turnaround sequence. Note that the boxes marked on the image relate to zones of interest. Each event detector is assigned a zone of interest and detects events by densely sampling local features from that zone throughout each sequence.

4 Evaluation

4.1 Events on the Airport Apron

We evaluate our method on a collection of 37 recordings of aircraft servicing turnarounds taken by a fixed camera on one apron at Toulouse airport. Each sequence is roughly an hour long recorded at 10 frames per second, starts a few minutes before an aircraft arrives and ends a few minutes after the aircraft departs. As such, each sequence wholly contains a complete aircraft servicing operation. We produced ground truth for the dataset by annotating start and end times for all event instances. In all experiments, we follow the leave-one-out protocol, training on 36 recordings and testing on the remaining one. The experiment is repeated with each sequence as the test sequence, and the results are averaged.

The event vocabulary comprises 12 servicing event classes which can take place on the apron; with the associated number of occurrences within the dataset given in Table 1. We observe that all events co-occur in at least one sequence, but for some event pairs the number of co-occurrences is as low as 3. There is a strict ordering on a small subset of these events (Aircraft Departure must occur after Aircraft Arrival) but in the main there is no particular sequence to the events; the processes are not directly linked but their timings are correlated. Some of the events on the apron are very simple. Events falling into this category would be the Aircraft Arrival/Departure, the Passenger Bridge Positioning and the Ground Power Unit Positioning, which all involve a single agent and a distinctive pattern of motion. In contrast are events such as Belt Loader Loading, which involve a variable number of participants and have greatly increased variability in duration and associated motion. An additional challenge in the dataset is the variable weather conditions: several turnarounds include rain and one takes place in snow. Lighting conditions also vary, with two turnarounds in the glare of the early morning sun, three taking place at sunset and one taking place after dark. These complicating factors impact on the performance of our detectors.

Event Class	Occurs	noT	RGMM	DG	DGMM	PGMM
Aircraft Arrival	37	98	**100**	99	**100**	**100**
Aircraft Departure	37	**100**	**100**	**100**	**100**	**100**
Passenger Bridge Positioning	37	98	**100**	97	**100**	96
Passenger Bridge Parking	37	90	92	98	**99**	**99**
Ground Power Unit Positioning	37	86	79	**96**	89	87
Handler Deposits Chocks	40	40	54	49	61	**63**
Container Front Door Loading	13	77	**80**	**80**	74	**80**
Container Front Door Unloading	14	77	79	83	81	**91**
Container Rear Door Loading	27	47	37	40	53	**58**
Container Rear Door Unloading	25	16	40	41	**56**	**56**
Belt Loader Loading	36	61	64	63	**71**	65
Belt Loader Unloading	25	71	78	71	78	**83**
Overall	365	73	77	78	**82**	**82**

Table 1: Numbers of occurrences of each event class in dataset and Average Precision of detection techniques.

4.2 Training Independent Event Detectors

We refer to results of [20] ,[11] as motivation for using Histograms of Optical Flow (HOF) as the features in our pipeline. Several of our events involve very slow and gradual movements, whilst in others objects accelerate rapidly making us wary of employing interest point detection for feature extraction. To handle the variation, it might be possible to tune interest point detectors for each event, but we simply use dense sampling in all cases and rely on our non-linear classifier to distinguish the features which are relevant. Since all events in our vocabulary must happen in designated areas, we are able to specify zones of interest relevant to each event and by using local features within these areas get the flexibility of local features whilst eliminating large amounts of noise. The zones used are depicted in Figure 3. Similarly, we fix the scale of the features for all event detectors to a block size of $3 \times 3 \times 2$, with a cell size of $18 \times 18 \times 8$ pixels. We validated our implementation by testing on the KTH actions dataset [17] and recorded results in line with [20].

We perform K-means clustering over 500,000 randomly sampled feature vectors to build a codebook of 4000 spatiotemporal words. Using this codebook, we convert extracted features to a 4000 dimensional histogram at each timestep in each zone. These vectors are then stacked through time in a matrices of size $4000 \times T$, summing cumulatively over the time dimension. This representation makes it possible to extract the bag of words for any interval with just one row subtraction. We train a binary SVM with probabilistic output using libsvm [3], for each event with a χ^2 kernel [10] on the ground truth samples and 10,000 negative samples (with mid point sampled randomly and duration sampled from a Gaussian distribution over duration for that event class).

At detection time, we pass histograms corresponding to every possible midpoint and all durations within 2 standard deviations of the mean for each event type to our trained classifiers, and the classifier for each event class supplies the probability of that interval containing an instance of that event.

We report the performance of the thresholded independent detectors in Table 1.

4.3 Results

The comparison of the independent detector with four variants of our temporal model can be found in Table 1. To be considered as true positives, we required detections to have a relative overlap (RO) > 30% with ground truth. Where RO between two intervals A,B is defined as $RO(A,B) = \frac{|A \cap B|}{|A \cup B|}$. We found that increasing the overlap level required had little effect on most events up to around the 60% mark, after which it degraded steadily. We believe this is due to our independent detectors distinguishing only the main significant motion associated with each event, and not necessarily the less distinctive but semantically important start and end points. The numbers given are Average Precision figures calculated from precision-recall graphs for each event. It was possible to obtain these naturally for the independent detector output by varying a threshold and taking non-overlapping local maxima; and in the case of the temporal models by varying the α_{c_i} variables in Equation 1.

There are five different sets of results in Table 1: 'noT' corresponds to the thresholded independent event detector output without a temporal model, 'DGMM' is our full model, 'RGMM' is the same model but with the tree structure determined randomly, 'PGMM' is again the same model but using a fixed tree structure determined in the manner of Pictoral Structure Models [6], 'DG' is the case where the pairwise probabilities are modelled by a single Gaussian, but we use our dynamic structure assignment.

Its clear that for almost all events, all variants of our temporal model perform significantly better than the 'noT' detector. Our observation that the single Gaussian was a poor approximation for the pairwise distributions has been validated with the GMM-based models performing better in almost all cases. There are a couple of exceptions to this; on the Container Front Door Loading/Unloading events the single Gaussian does well. It is likely no coincidence that these are the events with fewest occurrences in the dataset meaning that the pairwise distributions learnt in relation to them would have been trained on just a handful of examples. We put no special mechanism in place to deal with such cases; relying instead on our Bayesian GMM training method to avoid overfitting. The 'RGMM' results highlight the need for a good structure learning procedure when applying tree-shaped priors, displaying worse performance even than the 'DG' method. There is no clear winner from the other two connection strategies we tested, with both 'DGMM' and 'PGMM' performing consistently well suggesting that use of the data offered no advantage over a strong offline strategy in this case.

5 Conclusion

In this paper, we have introduced a novel method for exploiting temporal structure in scenarios to significantly improve over the performance of independent event detectors. The method is flexible; allowing event detectors to be trained independently and new/different detectors to be 'plugged in' as required. We presented an efficient algorithm for optimising the model and a novel, dynamic method of structure learning. In future work we would like to do further experiments on other datasets to fully determine the impact of dynamic structure learning. We are also interesting in exploring the possibility of extending our model to localize events in space as well as time, allowing its application in less constrained domains.

6 Acknowledgments

We thank colleagues in the Co-Friend project consortium (www.cofriend.net) for their valuable inputs to this research, and the EU Framework 7 for financial support (Co-friend FP7-ICT-214975).

References

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. ISBN 0387310738.

[2] M. Bregonzio, S. Gong, and T. Xiang. Recognising action as clouds of space-time interest points. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1948–1955. IEEE, 2009.

[3] C.C. Chang and C.J. Lin. LIBSVM: a library for support vector machines. 2001.

[4] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*, pages 65–72, 2005.

[5] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce. Automatic annotation of human actions in video. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1491–1498. IEEE, 2010.

[6] P.F. Felzenszwalb and D.P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, 2005. ISSN 0920-5691.

[7] S. Hongeng, R. Nevatia, and F. Bremond. Video-based event recognition: activity representation and probabilistic recognition methods. *Computer Vision and Image Understanding*, 96(2):129–162, 2004.

[8] Yuri A. Ivanov and Aaron F. Bobick. Recognition of visual activities and interactions by stochastic parsing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):852–872, 2000. ISSN 0162-8828. doi: http://doi.ieeecomputersociety.org/10.1109/34.868686.

[9] J.B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society*, 7(1):48–50, 1956.

[10] I. Laptev, B. Caputo, C. Schüldt, and T. Lindeberg. Local velocity-adapted motion events for spatio-temporal recognition. *Computer Vision and Image Understanding*, 108(3):207–229, 2007.

[11] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008*, pages 1–8, 2008.

[12] P. Matikainen, M. Hebert, and R. Sukthankar. Representing pairwise spatial and temporal relations for action recognition. *Computer Vision–ECCV 2010*, pages 508–521, 2010.

[13] J.C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *International Journal of Computer Vision*, 79(3):299–318, 2008.

[14] Juan Carlos Niebles, Chih-Wei Chen, , and Li Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *Proceedings of the 12th European Conference of Computer Vision (ECCV)*, Crete, Greece, September 2010.

[15] Nuria Oliver, Ashutosh Garg, and Eric Horvitz. Layered representations for learning and inferring office activity from multiple sensory channels. *Computer Vision and Image Understanding*, 96 (2):163 – 180, 2004. Special Issue on Event Detection in Video.

[16] C.E. Rasmussen. The infinite Gaussian mixture model. *Advances in neural information processing systems*, 12:554–560, 2000.

[17] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local SVM approach. In *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 3-Volume 03*, page 36. IEEE Computer Society, 2004.

[18] C. Stauffer and WEL Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–757, 2000.

[19] V.T. Vu, F. Brémond, and M. Thonnat. Automatic video interpretation: A novel algorithm for temporal scenario recognition. In *Proceedings of the 18th international joint conference on Artificial intelligence*, pages 1295–1300. Morgan Kaufmann Publishers Inc., 2003.

[20] H. Wang, M.M. Ullah, A. Klaser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *British Machine Vision Conference, London, UK*, pages 1–11, 2009.

[21] T. Xiang and S. Gong. Beyond tracking: Modelling activity and understanding behaviour. *International Journal of Computer Vision*, 67(1):21–51, 2006.