Autonomous Surveillance Tolerant to Interference

Nadeesha Oliver Ranasinghe and Wei-Min Shen

Information Sciences Institute, University of Southern California
{nadeesha,shen}@isi.edu

Abstract. Autonomous recognition of human activities from video streams is an important aspect of surveillance. A key challenge is to learn an appropriate representation or model of each activity. This paper presents a novel solution for recognizing a set of predefined actions in video streams of variable durations, even in the presence of interference, such as noise and gaps caused by occlusions or intermittent data loss. The most significant contribution of this solution is learning the number of states required to represent an action, in a short period of time, without exhaustive testing of all state spaces. It works by using Surprise-Based Learning (SBL) to reason on data (object tracks) provided by a vision module. SBL autonomously learns a set of rules which capture the essential information required to disambiguate each action. These rules are then grouped together to form states and a corresponding Markov chain which can detect actions with varying time duration. Several experiments on the publicly available visint.org video corpora have yielded favorable results.

Keywords: Machine Learning, Development Learning, Predictive Modeling, Recognition, Gap Filling, Temporal and Sequential Learning.

1 Introduction

There is a large amount of video data generated for surveillance of human activity. Traditionally, the surveillance of video data is performed by trained humans. However, this is becoming impractical due to the vast amount of data and the high potential for error. Humans are able to recognize the occurrence of specific actions such as approach, carry, walk, stop, etc. in these videos. Humans are also able to interpolate and predict what may have happened when an occlusion or gap appeared in the data stream. For example, if two people are observed approaching each other with two unique objects in their hands and later they are seen walking away with the objects interchanged, although the action of exchanging the objects may not have been seen, a human analyst can deduce that an exchange action had occurred. This paper demonstrates a system that is capable of performing such reasoning without any human intervention. This problem is challenging as even a small video contains a large amount of information from which the evidence that an action took place needs to be carefully extracted.

G. Herrmann et al. (Eds.): FIRA-TAROS 2012, LNAI 7429, pp. 73–83, 2012.

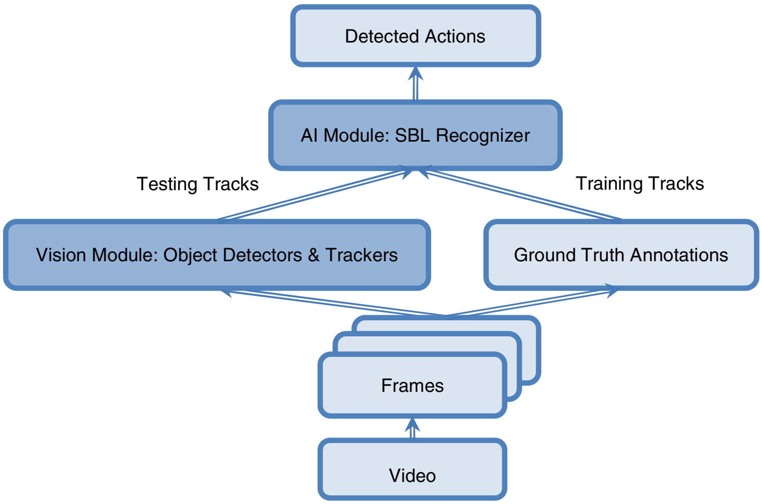


Fig. 1. Modular approach for action recognition

Since both computer vision and artificial intelligence are required for feasible solutions, we propose a modular approach as shown in Fig 1. The low-level computer vision module analyzes each frame of a video, extracts the relevant objects, tracks them across frames and then forwards these tracks to the high-level artificial intelligence module for action recognition. In this paper, we present the use of Surprise-Based Learning (SBL) [1-2] for the high-level module.

Our vision module is comprised of a specialized human detector [3], several Feltenzschwab object detectors [4], background subtraction and a fast moving blob detector. These detectors are fairly noisy, resulting in a reasonable amount of true and false tracks. Thus, SBL is responsible for filtering false tracks and using the true tracks to establish the likelihood that a particular action occurred. When training we assume the use of ground truth annotated data or synthetically generated data in order to eliminate false tracks.

One of the major challenges for this problem is learning an appropriate model for an action. Difficulties arise due to the continuous (non-discrete), uncertain and vast information space. Our approach to this is to perform simultaneous representation and learning. The learner learns a prediction model which forecasts the expected outcome of an action at any point in time. If the expected outcome does not match the observation, then this is known as a “surprise”. The algorithm attempts to identify the cause of each surprise and adapts its representation to minimize future surprises. During recognition, challenges arise from noise as well as variable durations of an action. To address this, SBL was augmented with Markov chains. Finally, there are challenges in evaluating the quality of the results, as some actions are ambiguously defined or biased based on the context of the scene depicted in the video. We address this to some extent by using at least two different criteria gathered from human evaluators, namely the precise number of votes and majority votes obtained per action per video.

The rest of this paper is organized as follows: Section 2 highlights related work. Section 3 describes a solution based on SBL, with details of the recognition system in section 4. Section 5 presents the experimental results and section 6 provides a discussion on this approach.

2 Related Work

Autonomous action recognition from sequential video data has been attempted with a variety of machine learning algorithms with varying levels of success. Some researchers have eliminated the overheads for higher level reasoning by reducing this problem to classification of features extracted from the data stream such as SIFT or HOG. An example of using Support Vector Machines [4] demonstrates that it is feasible. However, we are interested in high level reasoning approaches such that the recognition could be extended to handle gaps in the data and provide feedback to the low-level vision components (consider [5] as an example application).

Hidden Markov Models (HMM), Conditional Random Fields (CRF) and Neural Networks (NN) are some of the popular supervised learning algorithms that show some promise [5]. An example of using HMMs for action recognition is in [6-7], but as the number of hidden states is fixed the learner must search many possibilities prior to making a commitment for each action. CRFs share the same drawback as mentioned in [8]. The number of hidden layers in an NN is also empirically established, thus requiring a large amount of time and data for training. Most of these techniques require some extent of human intervention to design the approximation functions which map the sensors to states. In contrast, SBL is capable of learning the sensor mappings and adjusting the number of states autonomously. Its capability to learn a generic model of an action from a few examples, and perform recognition in the presence of noise will be investigated in this paper.

3 Surprise-Based Learning

The SBL algorithm is detailed in section IV of [1] for an autonomous robot. In this application the core algorithm was used without any changes. Hence, we will present the changes to the inputs and outputs of SBL required for autonomous video action recognition.

$$\text{Prediction Rule} \equiv \text{Conditions} \rightarrow \text{Predictions} \tag{1}$$

$$\text{Condition} \equiv (\text{Entity, Attribute, Comparison Operator, Value}) \tag{2}$$

$$\text{Prediction} \equiv (\text{Entity, Attribute, Expected Change, Value}) \tag{3}$$

SBL stores the learned knowledge for each action in a model, which consists of prediction rules as in (1). When an action is being executed in a video, the conditions (2) of a prediction rule are logic sentences describing the state of the observed entities and attributes at the current frame, while the predictions (3) are logic sentences that

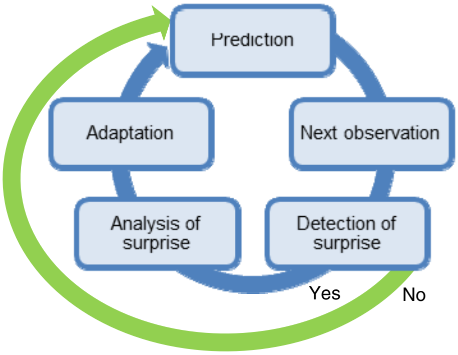


Fig. 2. Surprise-Based Learning process

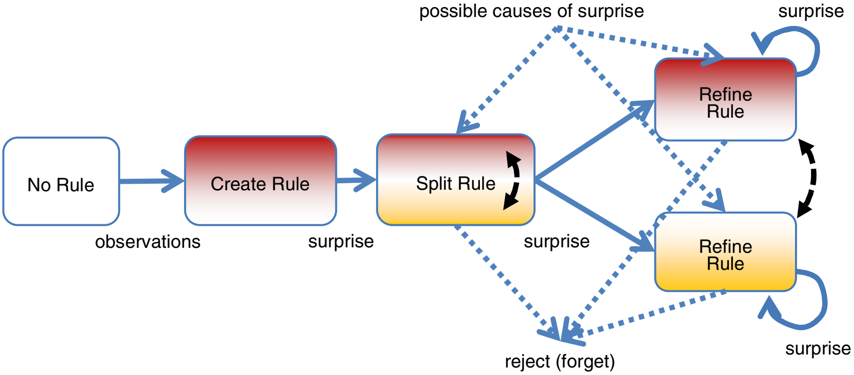


Fig. 3. Life cycle of a prediction rule

describe the expected change in the observed entities and attributes at the next sampled frame.

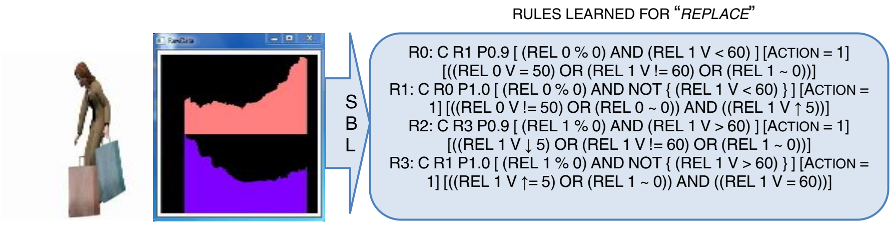
The learning process highlighted in Fig 2 maintains the prediction model as follows: i) The predictor returns all prediction rules in the model whose conditions match the current observation (e.g. frame k). ii) A new observation is made after a fixed amount of time has elapsed (e.g. frame k+t). iii) If no prediction was made or new learning opportunities exist, then new rules are created. iv) If surprise detection did not discover a discrepancy between the predicted (forecasted) observation and the current observation then make another next prediction and repeat the cycle, else v) surprise analysis returns the possible causes of surprise. Finally, vi) the model is revised through rule splitting and refinement to reflect the new observation as depicted in Fig 3.

In brief, new rules are created by comparing two subsequent observations with a set of predefined comparison operators. When a single rule is surprised for the first time it is split into two complementary rules by appending a possible cause from surprise analysis. One rule is specialized and the other is generalized to accurately capture the original and surprised results. All subsequent surprises to a complementary rule result in the pair being updated with new causes. When a pair of complementary rules describe a contradiction or encounter a number of consecutive surprises, they are forgotten or marked as inappropriate for future learning.

4 Action Recognition System

4.1 Learning Actions

To learn an action SBL is provided with a stream of data which typically includes the location of human actors and objects at each frame, some pre-processing is applied to compute their velocities across several frames and relative distances from each other. SBL will sample this data at fixed intervals such as every 10 frames with a set of standard comparison operators that allow it to detect changes including presence, absence, greater-than, less-than and equal-to.



R0: C R1 P0.9 [(REL 0 % 0) AND (REL 1 V < 60)] [ACTION = 1] [((REL 0 V = 50) OR (REL 1 V != 60) OR (REL 1 ~ 0))]
R1: C R0 P1.0 [(REL 0 % 0) AND NOT { (REL 1 V < 60) }] [ACTION = 1] [((REL 0 V != 50) OR (REL 0 ~ 0)) AND ((REL 1 V ↑ 5))]
R2: C R3 P0.9 [(REL 1 % 0) AND (REL 1 V > 60)] [ACTION = 1] [((REL 1 V ↓ 5) OR (REL 1 V != 60) OR (REL 1 ~ 0))]
R3: C R1 P1.0 [(REL 1 % 0) AND NOT { (REL 1 V > 60) }] [ACTION = 1] [((REL 1 V ↑= 5) OR (REL 1 ~ 0)) AND ((REL 1 V = 60))]

Fig. 4. Sensor data and prediction model for "replace"

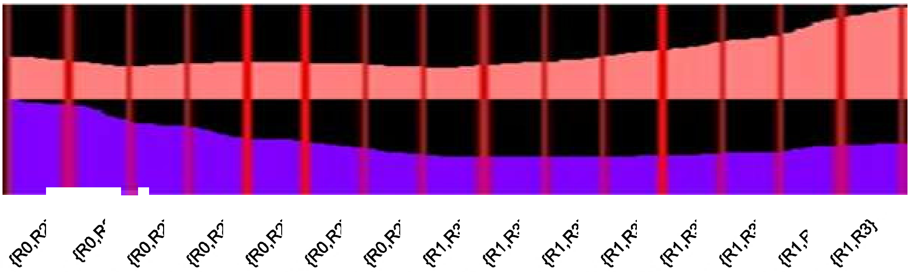


Fig. 5. Segmented data with active rules for "replace"

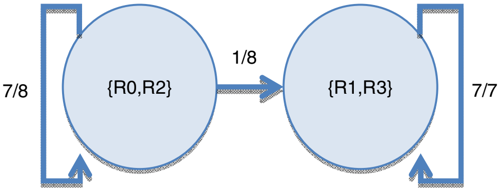


Fig. 6. Markov chain for "replace"

As an example consider the "replace" action shown in Fig 4. A human carrying a bag approached a stationary bag, replaced it with the one that she carried and walked away with the bag that was originally stationary. The image adjoining the human contains two graphs. The area marked in the lighter color (pink) indicates the relative distance between the human and the bag she was carrying while the area marked in the darker color (purple) indicates the relative distance between the human and the bag that was initially stationary. Each relative distance was mapped to an entity (e.g. REL 0, REL 1 in the prediction model) in SBL with its value as the attribute (e.g. V in the prediction model). SBL sampled the data and learned the prediction model shown beside the graph, which has 4 prediction rules R0, R1, R2 and R3, where R0 & R1are complementary and R2 & R3 are complementary.

Then, SBL will parse the data again with the prediction model to determine which rules are fired in each segment as shown in Fig 5. States are created by grouping the rules that fired simultaneously (e.g. S1 = {R0,R2} & S2 = {R1,R3}). The data stream can then be represented as a sequence of states e.g. $D_{replace} = [S_1, S_1, S_1, S_1, S_1, S_1, S_1, S_2, S_2, S_2, S_2, S_2, S_2, S_2, S_2]$. The transitions between these states are captured in a Markov chain as shown in Fig 6. The probabilities reflect the normalized number of state transitions observed in the state sequence.

During this learning or training phase SBL can be shown multiple examples possibly reflecting slightly different exemplars of a given action such that the prediction model is refined with each new video. However, once the Markov chain is created it is important to parse all the training videos again to accurately calculate the state transition probabilities. This process enables SBL to adjust the state space dynamically during learning as opposed some competitive techniques such as Hidden Markov Models and Neural Networks.

4.2 Recognizing Actions

When an unclassified video is presented to SBL, it will attempt to establish the similarity between the data and each trained action. This is performed by applying the data stream to the prediction model and extracting the rules that fired successfully at each sample segment. If a rule fired successfully then its conditions and predictions were satisfied by the data stream. Hence, the data stream can be converted to a state sequence by grouping these rules e.g. $D_{test} = [S_1, S_1, S_1, S_2, S_2]$.

For each learned Markov chain a similarity metric is calculated by validating the observed state sequence against it. This metric applies a predetermined negative penalty for each invalid start state, end state and state transition, while each valid state transition is scored zero. This metric allows SBL to compare videos of varying lengths. If multiple chains represent different exemplars of the same action, then sum their values to compute the metric for comparison.

A challenge lies in the fact that the entities stored in the prediction model may match several entities in the testing video. For example when testing the "replace" video in Fig 4 if there was a third bag in the scene then there would be 6 possible bindings to the two entities in the prediction model that need to be tested. As some of these mappings are interchangeable SBL still needs to verify at least 3 bindings. Therefore, the strategy is to establish the similarity for each unique binding by parsing video multiple times and keeping the best score (lowest value) as the most likely match.

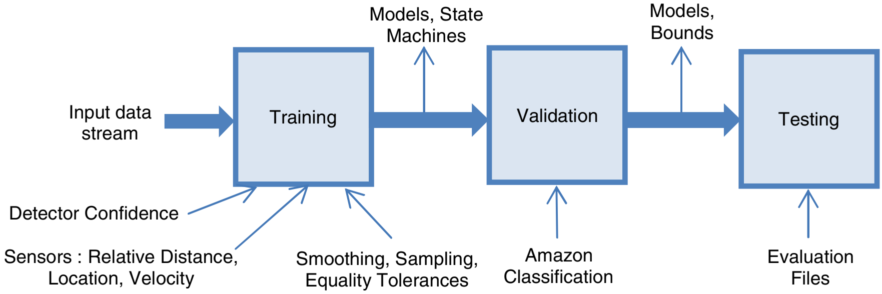


Fig. 7. SBL action recognizer inputs & outputs

Fig 7 depicts the overall action recognition process. The system is provided with a set of sensors to perform pre-processing on the data stream, and some parameters such as the cut-off level for the confidence of tracks, a smoothing rate to average the readings over a number of frames, the sampling frequency and the tolerance for equality of values. An advantage to this technique is that even with a short training phase comprised of only a few examples a reasonable prediction model and Markov chain can be learned. A validation phase is employed to improve the tolerance to noise. Here the system is shown all the videos that human evaluators have identified as the given action. It then calculates their similarity and stores the lower bound to identify the range of values that can be correctly classified as the trained action.

In the testing phase each video may contain multiple occurrences of an action. When computing the similarity the algorithm searches for matching pairs of start and end states as well as orphaned start or end states and reports them as separate detections of an action with the corresponding similarity value.

4.3 Noisy and Gapped Recognition

In a video data stream noise typically occurs due to temporary interference of sensors and inaccuracies in the detectors. Noise manifests as incorrect states in the sequence of observed states. Their short term nature results in small penalties on the similarity metric. SBL learns to tolerate noise by memorizing the acceptable range of similarity values from training data.

Gaps could occur in a video data stream due to temporary occlusions or sensor interference. If untreated these gaps would manifest as undefined states in the observed state sequence and produce very poor similarity values. SBL flags these gapped states and replaces them by performing a Breadth First Search originating from known states with the aid of the Markov chain. In particular, if the gap was at the beginning it would perform post diction from the first known state, if the gap was in the middle then it would perform interpolation between the known states and if the gap was at the end it would perform prediction from the last known state.

Once gap filling is performed slight penalties are applied to those states when calculating the similarity. This is to ensure that as more states are filled confidence in the action detected is decreased.

5 Results

The SBL recognition system was tested with several input streams to understand the quality of learning. Due to large computational costs of object tracking and limitations on the number of human annotators we selected several smaller datasets from the publicly available visint.org video corpora (http://www.visint.org/). Specifically ground truth annotations created by our USC group, ground truth annotations created by the VIGIL group (headed by the Stanford Research Institute), object tracks generated by the USC vision module, object tracks generated by the University of Purdue's vision module, and synthetic simulated data created by USC were used in several experiments. For brevity we will only present some of the key results here.

Synthetic or abstract training data was generated by observing the ground truth and vision module's data and abstracting it to remove noise in order to prevent over fitting. This was easily performed by visualizing an over smoothed output of the ground truth. It was observed experimentally that training with a few ground truth files or the equivalent synthetic data yielded high recognition success, while training from much noisier data generated by the vision modules tended to over constraint the prediction model and lower the recognition success. As with most learning algorithms, SBL demonstrated over fitting in the presence of noisy training data. For this reason, the results presented in this paper rely on synthetic data for training.

The first experiment was conducted using a dataset comprised of approximately 1200 videos for which USC object tracks were generated. SBL was tasked with recognizing 12 actions: approach, carry, catch, collide, haul, move, pickup, push, run, stop, throw and walk. SBL learned 38 prediction models as most of these actions had several different exemplars. Each video was about 1 minute in duration and contained at least 1 of the 12 selected actions. Excluding the vision module's processing time, SBL was able to classify 100 videos in approximately 1 hour.

The videos were broken into an overlap set which was used for validation & testing, and a novel set which was used for testing only. For each action the F1 score was computed with the formula: F1 = (2*precision*recall) / (precision+recall) where precision = (true positives) / (true positives + false positives) and recall = (true positives) / (true positives + false negatives). The positives and negatives for this calculation were obtained by requesting human evaluators (Amazon Mechanical Turks - AMT) to identify which actions they observed in each video. There was a noticeable amount of correlation error between the evaluators due to the ambiguity of some actions such as "run" and "walk", and biases introduced by the context of the scene. In order to understand the implications of these differences, the output was evaluated against two criteria, namely the exact vote per action per video and majority votes obtained across all exemplars of an action.

The results of SBL recognition based on this classification are presented in Fig 8. We see good F1 scores for approach, move, stop and walk which are all human centric actions, while actions that really on object detection such as catch, collide, haul, pickup, push and throw have lower scores. This behavior was expected as the USC vision module's human detector had much higher precision and recall than any other object detector. Given that there were many correlation errors and the majority of detectors had a precision of less than 20% the scores seen here are reasonable. This was validated by comparing the SBL results against results from hand-coded structured activity models that were also able to recognize these actions using the same USC object tracks. SBL outperformed the hand-coded models for each action and had an F1-score that was at least 10% higher. Notice that the disparity between the overlap set and novel set is very low in most cases indicating that SBL had learned relatively good models of each action.

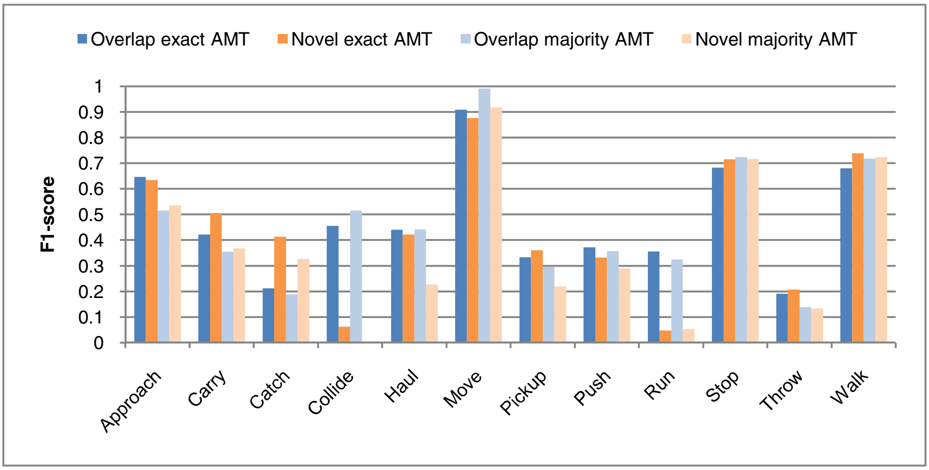


Fig. 8. Recognition results for the USC dataset

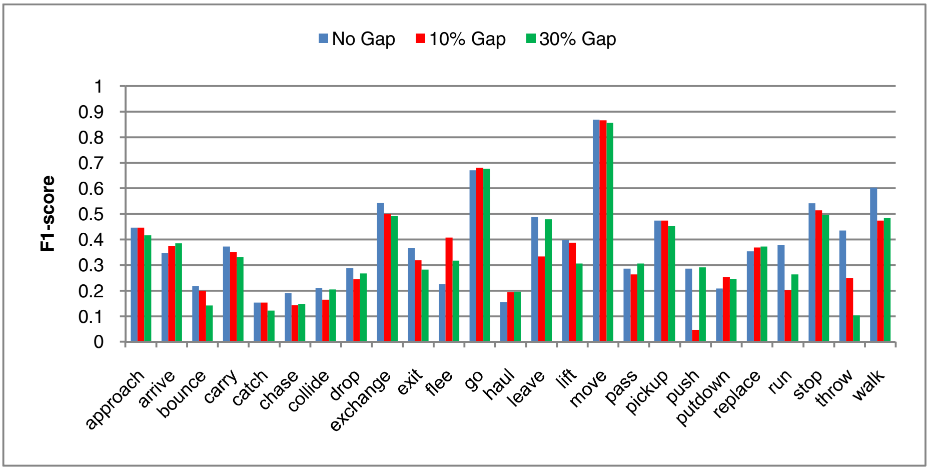


Fig. 9. Gap filling results for the VIGIL dataset

The second experiment was conducted using a dataset comprised of approximately 700 videos for which VIGIL ground truth annotations where available. SBL was tasked with recognizing 25 actions: approach, arrive, bounce, carry, catch, chase, collide, drop, exchange, exit, flee, go, haul, leave, lift, move, pass, pick up, push, put down, replace, run, stop, throw and walk. SBL learned 68 prediction models. The objective of this experiment was to introduce a large contiguous gap into each video at a random location and determine if SBL can correctly identify the action.

The results presented in Fig 9 show the recognition scores when there is no gap, a 10% gap and 30% gap respectively. The base case to compare is when there was no gap. A 10% gap meant that no data was available for 10% of the duration of each video, while the 30% gap meant that almost 1/3 of each video remained blank. The results confirm that SBL can indeed cope with gaps in the data stream. It is interesting to see that in some cases the gaps result in higher recognition by helping SBL reduce a number of false positives created by excessive noise in the detectors.

6 Conclusion and Future Work

This paper presented Surprise-Based Learning as a promising solution for autonomous recognition of a set of predefined actions in a stream of video data. SBL is robust to interference, such as noise and gaps caused by occlusions or intermittent data loss. Given a video, a low-level vision module extracted objects and presented their tracks to SBL. To learn an action SBL analyzed these data streams from a few example videos. For each exemplar of each action it built a prediction model and a corresponding Markov chain with the aid of several comparison operators. Then, as unclassified videos were presented SBL converted the data stream to states with the prediction model and computed the similarity against the Markov chain to establish if the action was present.

In practice learning from the video data stream resulted in models that were over fitting as SBL does not attempt to generalize. Instead, we used synthetic data generated by human observation of an action, resulting in fully autonomous recognition thereafter.

In future, there are several ways with which to improve the quality of the recognition. At present SBL only learns from positive examples of an action, yet it could discard false positives by learning from negative examples. This enhancement is currently in testing. Cleaning up the input data stream could help reduce the number of false positives and negatives. This could be done by improving the object detectors and through the use of pre-processors that incorporate some domain knowledge, such as filtering out objects and humans that are in the foreground and background using their scales, and ignoring objects that do not interact with the actors in the video.

In addition to detecting the action it is also important to establish the time at which it occurred. Currently, SBL provides this information and it will be evaluated in the near future. A major advantage of SBL in video surveillance is that it learns by detecting and analyzing surprises, which are analogous to anomalies. In particular, surprises taper off over a period of time meaning that an anomaly could become the norm, which is appropriate in situations such as people learning habits. Therefore, SBL is a highly adaptive learning technique that is also suitable for autonomous anomaly detection in the future.

Acknowledgements. We would like to thank our PI Prof Ramakant Nevatia, Prof Laurent Itti, Daniel Parks, Furqan Khan, Sung Chun Lee, Wei Jun and all the members of the USC computer vision and polymorphic robotics groups. This research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-10-2-0063. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

1. Ranasinghe, N., Shen, W.-M.: Autonomous Adaptation to Simultaneous Unexpected Changes in Modular Robots. In: Workshop on Recofingurable Modular Robotics at the International Conference on Intelligent Robots and Systems (October 2011)
2. Ranasinghe, N., Shen, W.-M.: Surprise-Based Developmental Learning and Experimental Results on Robots. In: International Conference on Development and Learning (June 2009)
3. Singh, V.K., Wu, B., Nevatia, R.: Pedestrian Tracking by Associating Tracklets using Detection Residuals. In: IEEE Motion and Video Computing (2008)
4. Felzenszwalb, P., Huttenlocher, D.: Efficient Graph-Based Image Segmentation. International Journal of Computer Vision (2004)
5. Bodor, R., Jackson, B., Papanikolopoulos, N.: Vision-based Human Tracking and Activity Recognition. In: Mediterranean Conference on Control and Automation (2003)
6. Qian, H., Mao, Y., Wang, H., Wang, Z.: On Video-based Human Action Classification by SVM Decision Tree. Intelligent Control and Automation, 385–390 (2010)
7. Diettrich, T.G.: Machine Learning for Sequential Data: A Review, Structural, Syntactic, and Statistical Pattern Recognition, pp. 15–30. Springer (2002)
8. Brand, M., Oliver, N., Pentland, A.: Coupled Hidden Markov Models for Complex Action Recognition. Computer Vision and Pattern Recognition, 994–999 (1997)
9. Weinland, D., Boyer, E., Ronfard, R.: Action Recognition from Arbitrary Views using 3D Exemplars. In: International Conference on Computer Vision, pp. 1–7 (2007)
10. Natarajan, P., Nevatia, R.: View and Scale Invariant Action Recognition Using Multiview Shape-Flow Models. In: International Conference on Computer Vision and Patten Recognition (2008)