

A framework for a video analysis tool for suspicious event detection

Gal Lavee · Latifur Khan · Bhavani Thuraisingham

Published online: 25 April 2007
© Springer Science + Business Media, LLC 2007

Abstract This paper proposes a framework to aid video analysts in detecting suspicious activity within the tremendous amounts of video data that exists in today's world of omnipresent surveillance video. Ideas and techniques for closing the semantic gap between low-level machine readable features of video data and high-level events seen by a human observer are discussed. An evaluation of the event classification and detection technique is presented and a future experiment to refine this technique is proposed. These experiments are used as a lead to a discussion on the most optimal machine learning algorithm to learn the event representation scheme proposed in this paper.

Keywords Video analysts · Semantic gap · Surveillance video · Unusual event · Event detection · Event classification · Event understanding

1 Introduction

Large quantities of video surveillance data exist in today's world. Cameras are everywhere constantly recording daily occurrences from many angles. The main use for this data is to review tape after a known event (e.g. a bank robbery) and gather information about that event (e.g. description of suspect) that will aid in bringing the offenders to justice. However, if the video analyst has no knowledge where and when or even if an event has

G. Lavee (✉)
Computer Science Department, Technion—Israel Institute of Technology,
Technion City, Haifa 32000, Israel
e-mail: gallavee@cs.technion.ac.il

L. Khan · B. Thuraisingham
The University of Texas at Dallas, 800 West Campbell Road, Richardson, TX 75080, USA

L. Khan
e-mail: lkhan@utdallas.edu

B. Thuraisingham
e-mail: bhavani.thuraisingham@utdallas.edu

occurred taking appropriate action becomes more difficult. To find an event such as this (if it exists) an analyst has to sort through many hours of video tape and make a judgment call whether the events pictured in the video require action to be taken. Let us consider the example of a corporate security guard monitoring surveillance video of a perimeter fence. If the guard suspects there may have been a breach of the perimeter fence at some point during the last 48 h, he would have to manually review 48 h of tape for each video camera recording a view of the fence to locate the breach. As a typical surveillance system may include tens or hundreds of camera angles, this quickly becomes a very daunting task.

This security guard (or other video analyst) could be helped a great deal if there existed a computer system which could learn events based on training examples and use this definition in conjunction with some configuration parameters to differentiate between “suspicious” and “normal” behavior. An interface to point out areas of particular interest in offline analysis could then be developed. At that time the analyst can exercise his judgment as to whether action needs to be taken based on this highlighted video segment.

Intrinsically tied to this work is the problem of bridging the semantic gap between the low-level features a machine sees when a video is input into it (e.g. color, texture, shape) and the high-level semantic concepts (or events) a human being sees when looking at a video clip (e.g. presentation, newscast, boxing match). Any approach to solving this problem must consider which low-level features will be extracted, how events should be represented and how unlabeled events will be classified. The first step towards solving this problem is to develop a robust event representation that characterizes the event (a series of frames in a video sequence) such that it can be meaningfully contrasted with other events. [6–10]

The event representation technique suggested in our paper is based on the paper by Zelnik-Manor et al. [9]. This method takes advantage of the fact that the same event captured by different camera configurations looks identical when projected on to the temporal dimension. Similar events which differ slightly in scale, translation, rotation and even color (different clothing) are classified as the same event. This characteristic allows us to cluster a video into event-specific segments, devise an event comparison method, and define “suspicious” activity by example.

We proposed and implemented a framework for a suspicious event detection system based on this technique. The design and implementation details of this framework are given in Section 2. A preliminary evaluation of this system’s event classification component’s performance is given in Section 3. In Section 4 we seek to determine an optimal learning algorithm for correctly classifying new events based on previously known manually (or automatically) labeled examples and propose an experiment to that end. In Section 5 we present our findings and discuss advantages and disadvantages of each of the proposed techniques. Distributed event detection is discussed in Section 6. Security and Privacy considerations are given in Section 7. In Section 8 we summarize the paper and discuss future directions.

2 Our approach

2.1 System design

Our proposed system takes a new unlabeled video sequence and multiple labeled video sequences representing different types of possible events. It produces a visualization of the content in the unlabeled video sequence as output to the user. The user can adapt this visualization according to their preferences (i.e. what type of event they consider to be suspicious) using the Video Analysis Tool interface. Figure 1 shows a block diagram of the system design.

The new video sequence is read in and stored as a matrix of RGB values over time ($\text{width} \times \text{height} \times 3 \times \text{number of frames}$). This phase can be thought of as the extraction of low-level features. An event representation (see below) for several overlapping sub-sequences is generated for use in event detection. These newly generated events are compared to a set of predefined events using the event comparison (distance) function defined below. The events are then classified by use of the nearest neighbor algorithm.

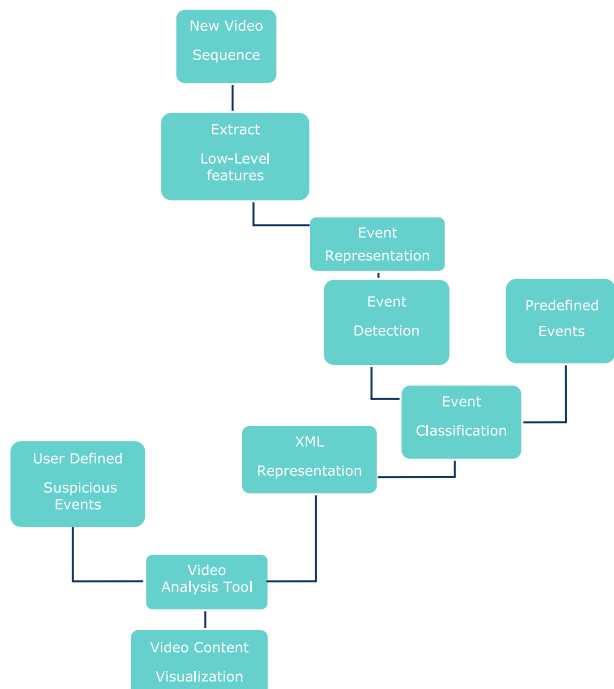
Once each of the overlapping events has a label the system generates a summary of events contained within the new video sequences and encodes it in an XML document.

This document is used in conjunction with user input to provide the appropriate visualization of the video content.

2.2 Event representation

This representation model effectively measures the quantity and type of changes occurring within a scene under the assumption that similar events (e.g. running, walking, waving) will have similar distributions of change for each of the three dimensions considered. A video event is represented as a set of x , y and t intensity gradient histograms over several temporal scales. Temporal scales are used to model the granularity of motion in an event (e.g. in a high-granularity temporal scale local motion like the swaying of the arms while walking will be captured while a low-granularity temporal scale will capture the overall motion of the subject across the frame). In our example we take four different temporal scales each with half the number of frames as the one preceding it. Temporal scale 1 is usually

Fig. 1 System design



composed of all frames in the video sequence. For example, if Video Sequence V1 is 120 frames long then the intensity value matrix of V1

- at temporal Scale 1 includes frames 1,2, ...,120 (120 frames)
- at temporal Scale 2 includes frames 1,3,5,...,120 (60 frames)
- at temporal Scale 3 includes frames 1,5,9,...,120 (30 frames)
- at temporal Scale 4 includes frames 1,9,17,...,120 (15 frames)

Once we obtain these 3D intensity value matrices for each temporal scale we are able to calculate the intensity gradient for each of the three directions.

Intensity gradient values measure the change in the intensity in each of the three directions (x,y,t) . Histograms of these values represent the distribution of change throughout the scene. A complete representation of an event is 12 of these 256 bin histograms (one for each of the (x,y,t) dimensions over four temporal scales). The gradient for each dimension is computed by averaging the difference between current value and next/previous value on the appropriate dimension. Figure 2 illustrates this process for a two-dimensional matrix.

2.3 Extracting the event representation

After reading in the video segment representing the event and extracting the 3D intensity matrix we compute the gradient of these values in the x , y and t directions for each of the four temporal scales. We then transform these values by taking their absolute value and normalizing the gradient vector to a length of 1. Before the histograms are computed (x,y,t) points where the t gradient is not above a certain threshold (these points are less relevant to the event) are taken out of consideration.

Histograms are computed by taking the range of these 12 normalized gradient matrices $(x,y,t \times 4$ temporal scales) and dividing it into 256 equal bins. A count of the gradient values that fall in each one of these bins is the histogram value at that bin's index. The y axis corresponds to the number of pixels whose gradient values fall into a particular bin and the x axis corresponds to the bin index. Each bin represents an equal interval of intensity gradient values. The cumulative interval of the bins spans the entire range of intensity gradient values. These histograms are, thus, fairly simple to plot, visualize and compare.

The histograms of longer length video segments will have larger magnitude peaks and thus a larger area. In order to different-length histograms to one another (to determine whether they represent the same event) we will need to normalize this area to equal 1.

After normalizing the area of the histogram the final step is smoothing the values. This is done to minimize the effect that local mismatches have on the comparison function. We do this by walking down the histogram and setting each bin value to the average value of the bins in its neighborhood (usually five bins on either side). While this method produces some artifacts such as a gentle slope towards the gradient extremes, it has proved to be effective in improving the comparison measure of important regions within the histograms.

2.4 Event comparison

Once we have a 12 histogram representation of a video segment we are tasked with comparing it to the representation of others to determine if the two represent similar high-

level semantic concepts (events). For this purpose a simple squared distance measure equation has been devised:

$$D^2 = \frac{1}{3L} \sum_{k,l,i} \frac{[h_{1k}^l(i) - h_{2k}^l(i)]^2}{h_{1k}^l(i) + h_{2k}^l(i)} \tag{1}$$

Where L is the number of temporal scales (four in our experiments) and $h(i)$ is the value of the histogram bin at index i . k represents the different dimensions (x,y,t) and l represents the four different temporal scales.

Fig. 2 Illustrates the calculation of the x and y gradient matrix from the original intensity value matrix a . Because our data is three dimensional we can extended this method to calculate the t gradient matrix as well

Matrix a

0	0	0	0
0	5	0	0
0	0	5	0
0	0	0	0

Produces the x direction

0	0	0	0
5	0	-2.5	0
0	2.5	0	-5
0	0	0	0

And the y direction

0	5	0	0
0	0	2.5	0
0	-2.5	0	0
0	0	-5	0

This measure averages the difference between each bin in the histogram of each dimension of each temporal scale over the total number of histograms. This produces a number that indicates how close the two compared events are to one another. The lower this number is the closer the two events are. A distance of zero results when an event is compared to itself.

This comparison function enables the use of the nearest neighbor classification algorithm to label an unknown video sequence with the label of the known event closest to it (in D^2 distance).

2.5 Event detection

Having a good event representation and event comparison function allows us to begin to detect and classify events in a new video sequence. However, the process determining which video frames constitute an event and how to separate adjacent events in a video sequence still requires definition. The event detection problem is, thus, defined in two different ways in this paper. The first is the simplified problem. In this problem we are given a video sequence that we know to contain a single event. We are to classify this event using our representation scheme and comparison function. In the second, more complex, interpretation of the problem we are given a video sequence containing an unknown number of events and are to determine which events occur and where (what frame numbers) within the video they occur.

The first step in our approach to both versions of this problem is generating a set of video segments (and their histogram representations) and manually classifying these. One or more of these “Classifiers,” as we have dubbed them, exists for each possible event in the scene. To improve accuracy multiple classifiers representing the same type of event should be different lengths, occur in different geographical regions of the scene and mirror each other’s direction of motion.

In the “simple,” one event per video segment, problem a new event is classified using the label of the classifier with the closest similarity to itself (the smallest distance value).

The broader problem, where an event can occur at any point within the video sequence, is a bit more challenging. To solve this problem we use a sliding window through the video sequence to capture fixed length events. In our experiment the window length was 25 frames with a skip interval of eight frames.. This means that each of the video windows created overlaps several other windows. A histogram event representation is generated for each of these windows and compared to the classifiers. Similar to the above problem each window is labeled according to the classifier it is most similar too. However, if this minimum distance is above a certain threshold (i.e. we cannot say this event is similar to any of the classifiers) the window is labeled as unknown. Each frame is then labeled with the mode classification of all the windows which contain it. This allows us to derive a frame by frame description/annotation of the events occurring within a given video segment. Figure 3 demonstrates this concept.

This paper explores this framework component further in later sections. We assess the performance of the techniques discussed above in Section 3 and propose an experiment to optimize efficiency in Section 4.

2.6 XML video annotation

Once a video sequence undergoes the event detection described in the previous section the events contained within it are stored in XML format. Because this format is easily machine readable the analyst is now able to sort through the video data much more efficiently (using



Fig. 3 Illustrates the method used to detect events in a video sequence with an unknown number of events. Fixed length overlapping windows are compared to “classifier” events. Their event type classification is then used to determine the event type of individual frames

the Video Analysis Tool). XML is also human-readable and thus manual viewing of event-content summary may also yield a good description of the video segment in question. The XML document contains a reference to the video data file, video segment specific attributes (such as video length and file format), and data on each of the events occurring within the video segment. Video events are modeled as complex elements and include such sub-elements as event type, event starting point and event duration.

This XML document annotation might on some future date be replaced by a more robust computer-understandable format (for example: the VEMML video event ontology language). This will enable a deeper level of event representation within video segments by making use of relationship cardinality, context definition and other features available within an ontology language.

Fig. 4 Screenshot of video analysis tool prototype



Table 1 Comparison of various “Disguised” event representation to the “classifier” events

	Disguise walking1	Disguise walking2	Disguise running1	Disguise running2	Disguise running3	Disguise waving1	Disguise waving2
Walking1	0.97653	0.948	1.411	1.3543	1.3049	13.646	12.792
Walking2	<i>0.45154</i>	<i>0.38097</i>	1.3841	1.1909	1.0021	13.113	12.132
Walking3	0.59608	0.53852	1.0637	1.0071	0.88092	13.452	12.681
Running1	1.5476	1.9412	<i>0.56724</i>	<i>0.61541</i>	<i>0.8114</i>	18.615	18.104
Running2	1.4633	1.844	0.97417	0.95833	1.1042	19.592	18.956
Running3	1.5724	1.8711	0.93587	0.94227	1.1189	18.621	18.029
Running4	1.5406	1.9673	1.0957	0.93731	1.0902	20.239	19.547
Waving2	12.225	10.191	11.629	13.141	12.801	<i>2.2451</i>	<i>3.1336</i>

Minimum distance is highlighted.

2.7 A video analysis tool

The availability of machine-readable annotation documents, whether these are in XML or a specialized video content ontology language format, is a big step towards the bridging of the semantic gap. A video analysis tool that takes this kind of annotation as input and organizes the corresponding video segment accordingly is certainly conceivable. This kind of utility could function as an aid to a surveillance analyst searching for “Suspicious” events within a stream of video data.

This activity of interest may be defined dynamically by the analyst during the running of the utility. The definition would then be compared to the video annotation and similar frames could be flagged for further analysis. Alternately, the analyst could define what is considered to be normal behavior and any annotated event deviating from this norm would be flagged.

The analyst would then consider all flagged areas to determine if action is required. A color coated scroll bar (with video segments of interest indicated by different colors) may be used as a graphical interface for this task. The analyst may then further refine the search parameters by marking false and true positives. Figure 4 shows a screen shot from a prototype of such a tool.

3 Preliminary evaluation of classification technique

We evaluated our classification technique with two different experiments. For both of these experiments we provided the system with manually labeled video clips (and their event representations) of three types (walking, running, and waving). These “classifier” video clips all portrayed the same actor on the same background (i.e. only the actions were variable). “Classifier” clips were of variable length, directionality of the event (e.g. left-to-right, right-to-left), and geographic location (e.g. waving occurred in different parts of the scene). We used Matlab [5] to generate the gradient intensity values over four temporal scales. From the resulting matrices we constructed the histogram representation for each event. We used the histogram distance function (Section 2) to compare event representations.

The first experimental problem we defined was to recognize and classify events irrespective of direction (right-to-left, left-to-right) and with reduced sensitivity to spatial variations (Clothing). The main assumption made in this experiment is that unlabeled video sequences are known to contain only one event. The unlabeled events used for testing the system showed the same actor performing the same type of events on the same background as

in the “classifier” videos, while wearing a different outfit or “disguise” to try and fool the system. A representation of each of these new “disguised” events is generated and compared to each of the classifiers using the comparison function described above. The new event is then classified using the label of the classifier with the closest similarity to itself (the smallest distance value). We contrasted this classification with the “truth” (manual labeling) of the “disguised” events to evaluate the accuracy of the classification scheme.

Table 1 shows the distance between each of the “disguised” events (columns) and the “classifier” events (rows). Highlighted cells indicate the minimum distance “classifier” event, whose label the system will use to label the “disguised” event.

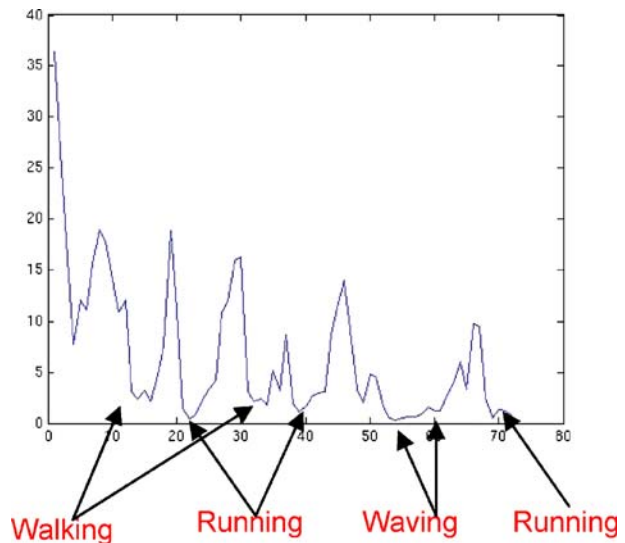
This method yielded 100% Precision (i.e. all disguised events were classified correctly). These results are, of course, not necessarily representative of the general event detection problem. Future evaluation with more event types, more varied data and a larger set of training and testing data is needed (Section 4).

The second experiment evaluated the scenario where an unlabeled video can contain any number of events. The goal of this experiment was to generate an accurate description of the high-level events within the unlabeled video. A sliding window was used to capture overlapping events of a fixed length (25 frames). Each event’s starting point was eight frames separated from the previous event.

Each of these window events is then classified in the same manner as the first experiment. The combined labeling of overlapping windows is used to determine the labeling for each video frame.

Figure 5 illustrates the minimum distance graph used to generate the video labeling. When watching the video sequence along with annotated event labeling it was observed to be similar to human perception. This is a subjective measure of performance and we have not yet developed a reliable metric for measuring the accuracy of such a video description. One possible option is to manually label each frame in the video sequence and compare this labeling to the automatic labeling frame by frame. This, however, is labor intensive and sensitive to the subjective perception of humans during the labeling.

Fig. 5 Illustrates the minimum distance graph used to generate event summary



4 The proposed experiment

We propose an experiment utilizing the event representation in [9] to determine the machine learning algorithm that will achieve optimal results in classifying unlabeled video segments based on the training data. As a preprocessing step we consider each of the 1,024 (256×4) histogram bins and their value as a feature, resulting in a 1,024 dimension feature vector (the traditional input for machine learning algorithms). We can further use techniques such as principal component analysis to reduce the dimensionality; however, in this experiment, we will use the raw feature vector.

4.1 The data

Each classifier will be fed a video sequence of length one to ten seconds featuring one of three events (walking, running, and waving). Approximately 20 video clips will be collected and used to test each algorithm. The event videos will each contain one actor against a (relatively) motionless background performing one of the three actions. The various actors in the event videos will be of different sizes, genders and ethnicities and will be wearing different clothing. Left to right as well as right to left examples of “directional” events (e.g. walking, running) will be included in the video data. Different backgrounds will also be used throughout the various clips. Video data will be collected using a Canon Z100 camcorder.

4.2 Methods

- Video data will be processed into four histogram representation using Matlab
- Histograms will be reduced into feature vector
- Feature vectors of 19 (out of 20) clips will be used to train classifier (leave one out cross validation)
- Feature vector of remaining clip will be used to test the classifier
- Output label produced by classifier will be compared against ground truth (manual labeling)
- Repeat these steps for each of the video clips
- Calculate precision of classifier (correct classifications/all examples)
- Calculate the confusion matrix of classifier (between events)

4.3 The classifiers

The machine learning algorithms to be evaluated as classifiers in this experiment are as follows:

- Nearest Neighbor Algorithm (Histogram Distance Function)—This is the classification algorithm proposed in [9] by Zelnik-Manor and Irani. A preliminary evaluation of its performance is detailed in the previous section.
- Nearest Neighbor Algorithm (Euclidean Distance)—This algorithm computes Euclidean distance between all feature vectors and selects the nearest Euclidean neighbor.
- Neural Networks—A network of perceptrons containing 1,024 inputs 1 output and multiple hidden layer nodes. Trained through the backpropagation algorithm.
- Decision Tree—This algorithm constructs a series of decision points based on features with minimal entropy.

5 Discussion

As is shown in Section 3 (albeit on a minimal dataset) the Nearest Neighbor classification algorithm using the histogram distance measure proposed in [9] performs very well in this kind of experiment. It will be interesting to see whether this performance level is maintained as the dataset size is increased and the training clips are more variable (i.e. different actors and backgrounds). Another interesting question is whether comparing histograms as feature vectors (rather than taking the average difference of corresponding bins) will degrade performance. Highly dimensional data can sometimes be problematic for decision tree algorithms, as individual features have only a small contribution to the classification of the data. Neural Networks, by contrast, have shown to be quite successful in recognizing patterns from high dimensional examples; however, it is not known if our limited dataset will suffice to properly train the network. Other classifying algorithms, not discussed in the proposed experiment such as support vector machines and Naïve Bayesian might also be used. However, for these to work some extra processing steps are necessary (defining a reliable probability distribution for Bayesian Networks and atomic concepts for support vector machines). The adaptation of the video data representation discussed in this paper for classification of video events using these methods is an interesting area for further exploration.

Another angle that might be further explored is the low-level feature input to our system. Texture and shape measures are most likely inapplicable as they consider image regions and we are mostly interested in pixel specific information. However, the multitude of color features available from different color-space mapping (e.g. HSV and Munsell space) to Opponent Color Axes could be examined to determine whether they have effect on classification accuracy. It will also be interesting to gauge the effect of Gaussian blurring filters or other computer vision techniques on the representation and accuracy of our system.

Refining the event classification component of our proposed framework for a Video Analysis Tool for Suspicious Event Detection will, increase efficiency and thus improve the productivity of the video analyst.

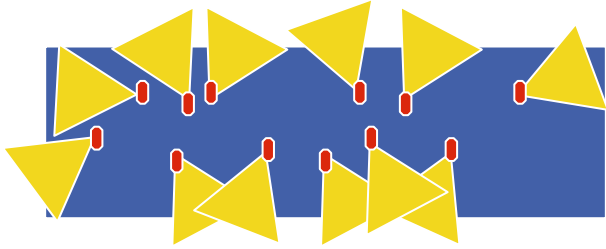
6 Distributed event detection

This paper focuses mostly on a single camera environment in which the data collection modeling and interpretation are all done. This is based on the assumption that one camera observes each scene. While this scenario is plausible in a real world surveillance scenario a more likely scenario would include several cameras watching over the scene from several different angles. The challenge now becomes adapting the system described in this paper to such a scenario [4].

In general we can say that if two camera are imaging the exact the same scene from different angles using the same camera configurations we can use the single camera training and classification methods on each of the cameras and combine the results or alternately we can generate the distribution based on the combined intensity gradient values from both scenes using a general “bucket of gradients” model.

In most cases, however, cameras will image only partially overlapping physical locations. This means that we will most likely need to change our system’s focus from analysis of video clips to analysis of specific location. That is, an analyst query will have to be location specific. For instance, instead of specifying all walking events as suspicious the

Fig. 6 A typical surveillance system. The imaged area (*yellow triangle*) of some security cameras may overlap. We must address the challenges caused by this (usually partial) overlap



analyst will have to specify all walking events in hallway no. 1 as suspicious. We can then use the mapping of all cameras imaging hallway no. 1 to analyze the events in the combination of their captured data. This kind of functionality will require adaptation of our video analysis, learning and classification techniques as well as enable mapping of our logical camera networks to physical locations they are imaging. Since a particular location may have different degrees of coverage by different camera angles a weighted contribution of these cameras to the event analysis will have to be considered. (Figure 6)

7 Security and privacy considerations

Discussion of data security usually centers on controlling access to the data with a focus on preventing access to authorized users. Video data objects, unlike relational data, contain information with poorly defined explicit content (semantic information). Because of this, while we can restrict access to video data based on its low-level features, timestamp, and (to some extent) physical location, the ability for true content based access control is still not attained. Clearly, this problem is quite similar to the video summarization problem and an effective technique for bridging the semantic gap, recognizing both semantic event and object content from low-level features and generating robust video descriptions would enable us to better define access control authorizations in terms of human understandable concepts.

Of course, a good access control model should be able to relate these high-level concepts to one another thus enabling minimal explicit authorizations in the policy base. This modeling can be made possible by making extensive use of hierarchical taxonomies describing event, object and location relationships, perhaps in combination with ontology language context and cardinality specifications. As techniques develop for extracting high-level concepts from low-level features the specification of these relationships will need to become more and more flexible and robust. [1, 3]

Privacy in the video data domain is not a well-defined concept (nor is it in other data domains). Work has been done concealing the identity of actors in the scene through pixilation or blacking out of portions of a frame. It would be interesting to carry out an experiment to determine whether such techniques would affect an event/behavior analysis such as the one proposed in Section 2 of this paper. Furthermore these techniques do not guarantee “privacy” (defined as protection of the actor’s identity) as inference techniques can be used to conjecture at the actor’s identity. Determining identity of people in video is a separate difficult problem and a topic of much research work in the nascent stages of its development. In other words, to preserve your privacy from an automated system the best technique is a false mustache. Manual identification is a tougher problem, especially as the

process of identification is much removed from the actual series of images displayed on a monitor. Like most privacy research we seek to find a delicate balance. In our case it is a balance between obscuring an actor's identity while still allowing the system to determine that actor's more general properties (e.g. detect him as a person). There is much work to be done in this area. [2]

8 Summary and direction

In this paper we proposed a framework for a video analysis tool for suspicious event detection. This tool is designed to reduce the demanding task of manually sorting through hours of surveillance video sequentially to determine if suspicious activity has occurred. We discussed the ideas behind the various components of this framework in detail as well as some of the implementation specifics.

We presented the results of some preliminary experiments we conducted to gauge the potential of this schema. The results of these experiments were promising but more in-depth analysis must be conducted before passing judgment on the representation, learning and classification schemes discussed in this paper. For this reason, we proposed an experiment that would contrast the performance of different learning algorithms and determine if we can improve upon the current technique. We provided some hypothesis on the results of such an experiment and contrast the strengths and weaknesses of the various algorithms.

We have also discussed future work that can be done building on the foundations laid out in this paper, including adaptation for distributed multi-camera environments and development of a complementary access control model to enable content based authorization policy.

In the continuation of this research path we will carry out the proposed experiments and report on the results and how they conform to our hypothesis. The addition of an access control component to restrict access to video content based on semantic events to our framework is another area that we are interested in exploring further.

References

1. Atluri V, Chun S (2004) An authorization model for geospatial data. *IEEE Trans Dependable Sec Comput* 1: 238–254
2. Bertino E et al (2000) An access control model for video database systems. In: *Conference on Information and Knowledge Management*, McLean, Virginia
3. Bertino E et al (2003) A hierarchical access control model for video database systems. *ACM Trans Inf Sys* 21:155–191
4. Chen T et al (2005) Computer vision workload analysis: case study of video surveillance systems. *Intel Technol J* 9(2);doi 10.1535/itj.0902
5. Natick MA (1992) *MathWorks, Matlab Reference Guide*
6. Rui Y, Anandan P (2000) Segmenting visual actions based on spatiotemporal motion patterns. In: *IEEE Conference on Computer Vision and Pattern Recognition*, June
7. Shi J, Malik J (1997) Normalized cuts and image segmentation. In: *IEEE Conference on Computer Vision and Pattern Recognition*. San Juan, Puerto Rico, June
8. Wang L, Khan L (2007) Automatic image annotation and retrieval using weighted feature selection. *Multimed Tools Appl* in press
9. Zelnik-Manor L, Irani M (2001) Event-based analysis of video. In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, December
10. Zhang H, Kankanali A, Smoliar W (1993) Automatic partitioning of full-motion video. *Multimedia Syst* 1:10–28



Gal Lavee graduated from the University of Texas at Dallas with a Master's Degree in Computer Science in December of 2005. During his time at UTD he worked in the Data and Applications Security lab and completed his Master's Thesis on "Suspicious" Event Detection in Video. He has published several conference papers including, most recently, an appearance in the proceedings of the 2006 ACM Symposium on Access Control Models and Technologies (SACMAT). Gal received his Bachelor's Degree in Computer Science from the University of North Texas in May of 2003. He is currently a PhD at the Technion in Haifa, Israel.



Dr Latifur R. Khan has been an Associate Professor of Computer Science department at University of Texas at Dallas (UTD) since September, 2006. He received his PhD and MS degree in Computer Science from University of Southern California (USC) in August 2000 and December 1996, respectively. He obtained his BSc degree in Computer Science and Engineering from Bangladesh University of Engineering and Technology, Dhaka, Bangladesh in November of 1993. After finishing his PhD, he joined at UTD as an assistant professor in September, 2000. Professor Khan is currently supported by grants from Air Force Office of Scientific Research, Raytheon, Nokia Research Center, Texas Instruments, Alcatel, USA and has been awarded the Sun Equipment Grant. Dr Khan has published more than 75 articles, book chapters, and conference papers focusing in the areas of: data mining, multimedia information management, and network security and intrusion detection. Professor Khan has also served as a referee for database/data mining journals, conferences (e.g., IEEE TKDE, KAIS, ADL, VLDB) and he is currently serving as a program committee member for Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (SIGKDD2006), and The 2006 IEEE International Conference on Data Mining (ICDM 2006). Dr Khan also gave a half day tutorial in 14th ACM International World Wide Web Conference, WWW2005, May 2005, Chiba, Japan and he has been an associate editor of Computer Standards and Interfaces Journal by Elsevier Publishing since June 2005.



Dr Bhavani Thuraisingham has recently joined The University of Texas at Dallas as a Professor of Computer Science and Director of the Cyber Security Research Center in the Erik Jonsson School of Engineering. She is a Fellow of the IEEE (Institute for Electrical and Electronics Engineers) and AAAS (American Association for the Advancement of Science). She received IEEE Computer Society's prestigious 1997 Technical Achievement Award for "outstanding and innovative contributions to secure data management." She was elected a Fellow of the British Computer Society in February 2005. Thuraisingham's research in information security and information management has resulted in over 70 journal articles, over 200 refereed conference papers, and three US patents. She is the author of seven books in data management, data mining and data security including one on data mining for counter-terrorism and another on Database and Applications Security. She has given over 25 keynote presentations at various research conferences and has also given invited talks at the White House Office of Science and Technology Policy and at the United Nations on Data Mining for counter-terrorism. She serves (or has served) on editorial boards of top research journals. Thuraisingham is also establishing the consulting company "BMT Security Consulting" specializing in Data and Applications Security consulting and training and is the Founding President of the company. Prior to joining the University of Texas at Dallas, Thuraisingham was an IPA (Intergovernmental Personnel Act) at the National Science Foundation from the MITRE Corporation. At NSF, she established the Data and Applications Security Program and co-founded the Cyber Trust theme and was involved in inter-agency activities in data mining for counter-terrorism. She has been at MITRE from January 1989 until June 2005 and has worked in MITRE's Information Security Center and was later a department head in Data and Information Management as well as Chief Scientist in Data Management. She has served as an expert consultant in information security and data management to the Department of Defense, the Department of Treasury and the Intelligence Community for over 10 years and is an instructor for AFCEA (Armed Forces Communication and Electronics Association) since 1998. Thuraisingham's industry experience includes six years of product design and development of CDCNET at Control Data Corporation and research, development and technology transfer at Honeywell Inc. Her academia experience includes visiting faculty at the New Mexico Institute of Technology, Adjunct Professor of Computer Science first at the University of Minnesota and later at Boston University. Thuraisingham was educated in the United Kingdom both at the University of Bristol and at the University of Wales.