Unusual Activity Analysis in Video Sequences

Ayesha Choudhary[1], Santanu Chaudhury[2], and Subhashis Banerjee[1]

[1] Department of Computer Science and Engineering
Indian Institute of Technology Delhi, New Delhi, India.
{ayesha,suban}@cse.iitd.ernet.in
[2] Department of Electrical Engineering
Indian Institute of Technology Delhi, New Delhi, India.
schaudhury@gmail.com

Abstract. We present a unique representation scheme for events in an area under surveillance, which provides a mechanism to analyze videos from multiple perspectives for unusual activity analysis. We propose clustering in event component spaces and define algebraic operations on these clusters to find co-occurrences of event components. A *usualness* measure for clusters is proposed that not only gives a measure on how usual or unusual an activity is, but also a basis for analyzing and predicting the possibly usual or unusual activities that can occur in the surveillance region.

Key words: Clustering, Unsupervised Learning, Unusual Activity Analysis, Event Recognition

1 Introduction

Automatic learning and detection of anomalous behavior from video sequences is an important area of research in computer vision, specially in the context of visual surveillance. Machine learning and probabilistic techniques are widely applied in this area. Most of the activity recognition systems predefine and model the anomalous activities so that the system can recognize whether the activities detected are anomalous or not [1]. Others learn the usual activity patterns either in supervised or unsupervised manner and then recognize unusual activities based on their dissimilarity from the usual ones. Supervised learning based methods not only need large volumes of training data, usually difficult to get for real world applications, they also suffer from the shortcoming that all activities in the real world cannot be predefined.

Given a long video sequence and no prior information of the scene, we propose a representation scheme for events that logically partitions the event feature vector. This representation allows us to apply different similarity measures on each of the components and cluster the event components rather than clustering the monolithic event vector. Therefore, it can be used for both video mining for similar events as well as unusual activity analysis. We propose a *usualness* measure on clusters that depends on the size of the cluster. As unusual activities

are rare and dissimilar from normal, clusters with low *usualness* measure depict unusual activities. The novelty of our work also comprises of the algebraic operations defined on these clusters, which along with the *usualness* measure associated with each cluster allows us to explore the space of all clusters for detecting unusual events in the video. It gives us a tool for finding co-occurrences of event components, thus, allowing analysis of the video from multiple perspectives. Moreover, these algebraic operations on the clusters of event components allow us to get back clusters of the monolithic event vectors. Therefore, there is no loss of information by clustering in the event component spaces instead of clustering in the event space. The co-occurrence calculus for clusters is not present in the literature and therefore, is unique and a novel contribution of our work. This event representation and clustering scheme can also be used to develop applications like Intelligent Fast Forward [2], where given a event segment in a video the system is able to move to the next or all portions of the video where a similar event occurs.

In the next section, we discuss some of the main techniques that have been applied for activity recognition. In section 3, we present the event representation scheme. Section 4 defines the clustering framework. In section 5, we present the results and conclude in section 6.

2 Related Work

As mentioned above, most activity recognition systems model and learn known activities. The Hidden Markov Models (HMMs) and its variants are most widely used for this purpose, [3],[4], [5], [6], [7], [8], [9]. HMMs are used by Starner and Pentland [6] for modeling hand gestures. Variants of HMMs, Parameterized-HMM (PHMM) [10], Coupled-HMM(CHMM) [7] are used for recognizing complex activities like interaction between moving objects in the scene. In [11] stochastic context-free grammar is used for computing the probability of temporally consistent sequences of primitive actions that are recognized by a HMM model. In [12], a model of stochastic context-free grammar is proposed for recognizing semantically meaningful behavior over extended periods. The authors in [13] propose *propagation networks* for modeling temporal inter-leavings of low level events which may occur concurrently in multi-object activities. Bayesian networks is yet another popular technique used for activity recognition [14], [15], [1], [16], [4]. In [1], [16], multi-layered FSM model is proposed for activity recognition where supervised training using Bayesian formulation is used for estimation of the parameters of their model. In [17], a multi-layered FSM framework is used for unsupervised learning of usual activities. In this method those activities that are not recognized as usual are flagged as unusual. Usual activities are learnt using unsupervised clustering in [18]. Unlike our approach, these two methods learn usual activity patterns for detection of unusual activities. In our approach, we cluster all events and based on their *usualness* measure, events are flagged as usual or unusual.

3 Event Representation

Events in a long video sequence are characterized by the position of moving objects, through time. In general, it is observed that objects tend to move from one landmark to another. These landmarks include locations from which objects enter the scene, exit the scene and in general, locations where they stand and wait. In our terminology, these landmarks are referred to as *attractors* and a trajectory is then an *attractor* sequence.

Thus, an event feature vector is a high-dimensional vector that contains low-level information about the object in the scene, its positions through time and the time during which it is visible in the scene. This leads to the problem of clustering heterogeneous data in high dimensional vector space. Clusters in this space give a restricted view of similarity of events. For example, if a person P_i traverses a landmark sequence LS_j during a certain time interval and another person P_j traverses LS_j during another time interval, the event vectors will be dissimilar and shall not be clustered together. Thus, even if it is common for an object of category *individual* to traverse landmark sequence LS_j, clustering in the high dimensional event space leads to the loss of this information. We represent an event as a tuple,

$$T_i = (OID, OC, LS, TI)$$

where,

- OID: *Object ID* is the ID given to an object when it enters the scene.
- OC: *Object Category* is the category to which the object belongs, for example, individual or group.
- LS: *Landmark Sequence* is the sequence of *attractors* that the object visits during its presence in the scene.
- TI: *Time Interval* is the time during which the person is visible in the scene.

This representation logically partitions the event vector into semantically meaningful quantities. Each component is of a different data type, not necessarily numerical, and the components are not comparable among themselves. Therefore, different similarity measures can be applied on each component and clustering can be done in the component spaces instead of the event space.

4 Clustering Framework

We define the similarity measure for tuples and the *usualness* measure for clusters below:

Definition 1 : *Similarity measure for tuples.* Assume that the data consists of tuples of the form $T_i = (t_{1_i}, t_{2_i}, \ldots, t_{m_i})$ where each component t_k represents a numeric or semantic data type. The components t_k's for all k need not be comparable among themselves. Let S_i be the similarity measure for the i^{th} component, t_i. Then, $S = (S_1, S_2, \ldots, S_m)$ defines the similarity measure between tuples T_i and T_j such that, $S(T_i) = T_j$ iff $S_1(t_{1_i}) = t_{1_j}, S_2(t_{2_i}) = t_{2_j}, \ldots, S_m(t_{m_i}) = t_{m_j}$. This similarity function defines an equivalence relation on the tuples.

Definition 2 : *Size of a cluster.* The number of items belonging to a cluster defines the size of the cluster.

Definition 3 : *Usualness measure associated with a cluster.* Let Ω be the set of all clusters, and $C \subset \Omega$ be a cluster of size x. The *usualness* measure function for a cluster C is defined as:

$$p(C) = \begin{cases} 0 & x < Thres_1 \\ e^{-(x-Thres_2)^2/(2*\sigma^2)} & Thres_1 \leq x \leq Thres_2 \\ 1 & x > Thres_2 \end{cases} \quad (1)$$

where,
$\sigma = (Thres_2 - Thres_1)/3$, $Thres_1$ and $Thres_2$ are thresholds on the rate of growth of the *usualness* of a cluster.

A cluster represents an unusual activity if this measure is 0. If the measure is 1, the cluster represents a usual activity. All values of $p(C) \in (0, 1)$, denote the extent to which the cluster represents a usual phenomenon. This is similar to the membership function defined for a fuzzy set.

4.1 Clustering in component spaces

The clustering algorithm is a dynamic incremental clustering algorithm, which is applied to each component of the event tuple that is formed as the video is parsed. As the clusters are created, the values of the other components for that event vector are also stored. A component, denoted by t, is clustered as follows:

- Let Ω be the set of all clusters of a particular component. Initially, $\Omega = \phi$, the empty set.
- When the first tuple is encountered, create a cluster C_1, and assign t_1 to it. $p(C_1) = 0$.
- As the tuples are encountered, two possibilities exist:
  - If $S(t_k) = t_i \in C_i$, assign t_k to cluster C_i and update $p(C_i)$.
  - Otherwise, create a cluster C_k and assign t_k to it. $p(C_k) = 0$.
- Repeat until all the tuples are clustered.

Thus, event components can be clustered without knowing the number of clusters *a priori* and clusters for each component depict how usual the occurrence of that component is. For example, in an airport the sequence of entering the airport and directly going to the airline desk is a commonly taken path depicting a usual event, whereas a person going from the entrance to a restricted area is a rarely traversed path depicting an unusual event. Thus, clustering in the event component space gives a flexible tool to evaluate the usualness of an event component without explicitly knowing which events occurred.

4.2 Properties of the *usualness* measure

The *usualness* measure defined on the clusters satisfy the following properties:

- $0 \leq p(C) \leq 1$
- $p(\phi) = 0$
- $p(A \cup B) = max\{p(A), p(B)\}$
- $p(A \cap B) \leq min\{p(A), p(B)\}$

where C is any cluster, ϕ is an empty cluster, and A and B are clusters either from the same or different component spaces. The union of two clusters defines the *OR* operation and is well defined if both the clusters belong to the same component space. It defines a commutative monoid on the space of all clusters.

4.3 Composition of Clusters

When the event component clusters are created or updated, if the tuple information is also stored, then composition of clusters give an insight into the co-occurrence of two or more event components. Let C_{x^*} be the cluster for the value x^* of the first component of the event cluster and C_{y^*} be the cluster for the value y^* of the second component of the event cluster. Then, a composition of the clusters will be the set

$$C_{x^* \otimes y^*} = \{(x_i, y_j) | S_x(x_i) = x^*, (x_i, y^*) \in C_{y^*} \text{and} S_y(y_j) = y^*, (x^*, y_j) \in C_{x^*}\}$$

where, S_x and S_y are the similarity measures on the x and y components.

Thus, when the values of the complete tuples are stored while clustering in the component space, the composition operation gets back the cluster in a higher dimensional space. This gives a powerful mechanism for getting all the cluster combinations in higher dimensional spaces from one-dimensional clusters. The *usualness* measure of the composite cluster can then be computed from its size.

Composition of clusters across spaces provides a tool to find the *usualness* measure of co-occurrence of two component values. For example, it answers queries of the form "Is it usual that groups of people traverse landmark sequence LS_1, from the entrance of the airport to the airline desk?" While the clusters in each component space provide only the knowledge of which component value occurs often, the composition of clusters gives us a different perspective to the state of the usual and unusual activities in the system.

In relational databases, a join operation combines records from two or more tables. The composition of clusters can be seen as a join operation between clusters, instead of records. This technique of manipulating the clusters gives us an insight into the state of the system. Moreover, the bounds on the *usualness* measure of the resulting clusters gives us an idea of the usualness of the co-occurrence of two components.

In case, it is desired to find the *usualness* measure of the composition of C_x and C_y, without considering whether the (x, y) tuple actually occurred as an event component, equation 2 gives the greatest lower bound and the least

upper bound on the *usualness* measure of the set $C_{x^*} \cap C_{y^*} = \{(x_i, y_j) | (x^*, y_j) \in C_{x^*} \text{and} (x_i, y^*) \in C_{y^*}\}$

$$p(C_{x^* \otimes y^*}) \leq p(C_{x^*} \cap C_{y^*}) \leq min\{p(C_{x^*}), p(C_{y^*})\} \tag{2}$$

These bounds can be used to find the *usualness* of the event tuples, for event components that may not have co-occurred. Thus, this also gives an insight into the usualness of events that may occur in the scene.

The composition of clusters, from all four component spaces, gives back the usualness of the event tuple in the database. For instance, if an event tuple $T = (P_1, P, LS_1, TI_1)$ occurred n times. Suppose that P_1 belonging to the category P traversed through landmark sequence LS_1 many times later in the video. Thus, the clusters for P_1, P, LS_1 each have size $>$ n, while TI_1 has size n. Then,

$$p(C_T) = p(C_{P_1 \otimes P \otimes LS_1 \otimes TI_1}) \leq min\{p(P_1), p(P), p(LS_1), p(TI_1)\} = p(TI_1)$$

This shows that the composition operation gives back the actual *usualness* measure of an event.

Therefore, properties of the *usualness* measure allow us to define well-defined operations on the clusters. Without explicitly storing the clusters in different dimensions, the composition operation gives back the clusters and their true *usualness* measure. This allows the user to get the information required for analyzing the activities as well as predicting the possibly unusual events that can occur in the area under surveillance. Thus, our event representation technique is powerful enough to give a multi-perspective view of the usual and unusual events in the scene as well as to find similar events across a long video sequence.

5 Results

In our implementation, adaptive background subtraction is used for detecting moving objects in the scene and estimating the category to which the object belongs. *Landmark Sequences* are found by finding the *attractors* at which the object enters the scene and the *attractors* it visits while it is in the scene. Finally, when the object exits from the scene, we cluster the event components. We use equality of components as the similarity measure. Our input is a long video of people walking in a long corridor of a building. The attractors are the entrances to the corridor and the doors of the various offices. Figure 1 shows frames taken from the result video. Figure 2 shows a log of the usual and unusual landmark sequences in the input video, which are consistent with the ground truth. The log in figure 3 shows composition of clusters for event components: object category and landmark sequences.

6 Conclusion

We proposed an event representation scheme where each component of the event vector is a logical entity. We cluster in the component spaces instead of clustering

Fig. 1. Frames from the input sequence: (a) Frame 530 (b) Frame 6669

the monolithic event vector. The proposed *usualness* measure on the clusters along with the algebraic operations defined on these clusters provide a flexible and well-defined tool to predict the co-occurrences as well as usualness of events. This method can be used for a variety of video applications, including unusual activity analysis and indexing and mining of videos.

```
File  Edit  View  Terminal  Tabs  Help
The landmark sequence is:
Attractor: 4
Attractor: 9
Attractor: 1
This is a usual landmark sequence for the scene

The landmark sequence is:
Attractor: 5
Attractor: 1
This is a usual landmark sequence for the scene

The landmark sequence is:
Attractor: 6
Attractor: 9
This is an unusual landmark sequence for the scene
The details of the event are:
The object id is: P11
The object category is: Person
The frame interval is from frame no: 9048 to frame no: 9134

The landmark sequence is:
Attractor: 5
Attractor: 6
Attractor: 9
Attractor: 1
This is an unusual landmark sequence for the scene
The details of the event are:
The object id is: P12
The object category is: Person
The frame interval is from frame no: 13808 to frame no: 13901

The landmark sequence is:
Attractor: 1
Attractor: 5
```

Fig. 2. The log of the video after clustering in the landmark sequence space.

```
File  Edit  View  Terminal  Tabs  Help
The co-occurrence of
Object Category: Person
 and
Landmark Sequence:
Attractor: 1
Attractor: 4
is 0.324652 usual.

The co-occurrence of
Object Category: Person
 and
Landmark Sequence:
Attractor: 9
Attractor: 4
 is 0.324652 usual.

The co-occurrence of
Object Category: Person
 and
Landmark Sequence:
Attractor: 4
Attractor: 9
Attractor: 1
 is usual

The co-occurrence of
Object Category: Person
 and
Landmark Sequence:
Attractor: 5
Attractor: 1
 is usual

The co-occurrence of
```

Fig. 3. The log for co-occurrence of event components.

References

1. Hongeng, S., Bremond, F., Nevatia, R.: Representation and optimal recognition of human activities. In IEEE Conference on Computer Vision and Pattern Recognition (2000) 1818–1825
2. Zelnik-Manor, L., Irani, M.: Event-based analysis of video. In IEEE Conference on Computer Vision and Pattern Recognition (2001) 123–130
3. Kettnaker, V.: Time-dependent HMMs for visual intrusion detection. In IEEE Workshop on Event Mining: Detection and Recognition of Events in Video (2003)
4. Medioni, G., Cohen, I., Bremond, F., Hongeng, S., Nevatia, R.: Event detection and analysis from video stream. IEEE Trans. on Pattern Analysis and Machine Intelligence, 23(8) (2001) 873–889
5. Moore, D., Essa, I., Hayes, M.: Exploiting human actions and object context for recognition tasks. In International Conference on Computer Vision (1999) 80–86
6. Starner, T., Pentland, A.: Real-time american sign language recognition from video using hidden Markov models. In SCV (1995) 265–270
7. Brand, M., Oliver, N., Pentland, A.: Coupled hidden Markov models for complex action recognition. In IEEE Conference on Computer Vision and Pattern Recognition (1997) 994–999
8. Oliver, N., Rosario, B., Pentland, A.: A Bayesian computer vision system for modeling human interactions. In International Conference on Computer Vision Systems (1999) 255–272
9. Yamato, J., Ohya, J., Ishii, K.: Recognizing human action in time-sequential images using hidden Markov model. In IEEE Conference on Computer Vision and Pattern Recognition (1992) 379–385
10. Wilson, A., Bobick, A.: Recognition and interpretation of parametric gesture. In International Conference on Computer Vision (1996) 329–336
11. Ivanov, Y., Bobick, A.: Recognition of visual activities and interactions by stochastic parsing. IEEE Trans. on Pattern Analysis and Machine Intelligence, 22(8) (2000) 852–872
12. Moore, D., Essa, I.: Recognizing multitasked activities from video using stochastic context-free grammar. In AAAI (2002)
13. Shi, Y., Bobick, A.: Representation and recognition of activity using propagation nets. In 16th International Conference on Vision Interface (2003)
14. Buxton, H., Gong, S.: Advanced visual surveillance using Bayesian networks. In International Conference on Computer Vision (1995) 111–123
15. Madabhushi, A., Aggarwal, J.: A Bayesian approach to human activity recognition. In 2nd International Workshop on Visual Surveillance (1999) 25–30
16. Hongeng, S., Nevatia, R.: Multi-agent event recognition. In International Conference on Computer Vision (2001) 84–93
17. Mahajan, D., Kwatra, N., Jain, S., Kalra, P., Banerjee, S.: A framework for activity recognition and detection of unusual activities. In Indian Conference on Computer Vision, Graphics and Image Processing (2004)
18. Zhong, H., Shi, J., Visontai, M.: Detecting unusual activity in video. In IEEE Conference on Computer Vision and Pattern Recognition (2004) 819–826