Group Event Detection for Video Surveillance

Weiyao Lin*, Ming-Ting Sun*, Radha Poovendran* and Zhengyou Zhang**

*Department of Electrical Engineering
University of Washington, Seattle, USA
{wylin, mts, rp3} @u.washington.edu

**Microsoft Research,
Microsoft Coop., Redmond, USA
zhang@microsoft.com

Abstract—This paper presents a novel approach for automatic recognition of group activities for video surveillance applications. We propose to use a group representative to handle the recognition with flexible or varying number of group members, and use an Asynchronous Hidden Markov Model (AHMM) to model the relationship between two people. Furthermore, we propose a group activity detection algorithm which can handle symmetric and asymmetric group activities, and demonstrate that this approach enables the detection of hierarchical interactions between people. Experimental results show the effectiveness of our approach.

I. INTRODUCTION

Detecting human group behavior or human interactions has attracted increasing research interests [1-6]. Some example group events of interests include people fighting, people walking together, people being followed, group conversations in a party, terrorist launching attacks in groups, etc. In this paper, we address the following issues for group event detection.

A. Group Event Detection with flexible or varying number of group members. Most previous group event detection researches [1-2] use a Hidden Markov Model (HMM) or its variation to model the human interactions. Some people try to recognize human interactions based on a content-independent semantic set [3-4]. However, most of these works are designed to recognize group activities with a fixed number of group members, where the input feature vector length is fixed. They cannot handle cases where the number of group members is flexible or even varying, which is often the case in our daily life (e.g., people may leave or join a group activity). In this case, the input feature vector length may vary with different number of group members. Although some works [5-6] tried to deal with the detection of group activities with varying number of members, most of them have specific assumptions that restrict their applications.

B. Group Event Detection with a Hierarchical Activity Structure. In many scenarios, interacting people form subgroups. However, these subgroups are not independent to each other and they may further interact to form a *hierarchical structure*. For example, in Fig. 1(a), three people fighting form a subgroup of *fighting* (the dashed circle). At the same time, another person is approaching the three fighting people and these four people form a larger group of *approaching* (the solid circle in Fig 1 (a)). This is an example of hierarchical activity structure with the group of *approaching* at a higher level than the group of *fighting*. Some algorithms [1-2] could be extended to deal with the problem of hierarchical structure event detection when the number of group members is fixed. However, to the best of our knowledge, our work is the first to address the problem of group event detection with *a varying number of group members* under a *hierarchical activity structure*.

C. Clustering with an Asymmetric Distance Metric. Most previous clustering algorithms [6,10] perform clustering based on a symmetric distance metric (i.e. the distance between two people is symmetric regardless of the relationship of the people). In the group event detection, some activities such as "following" are asymmetric (e.g. *person A following person B* is not the same as *person B following person A*). Defining a suitable asymmetric distance metric and performing clustering under the asymmetric distance metric is an important issue.

The contributions of this paper are summarized as follows: (1) To address the problem of detection with a hierarchical activity structure, we propose a *Symmetric-Asymmetric Activity Structure (SAAS)*. (2) To address the problem of detecting events with varying number of people, we propose to use a *Group Representative (GR)* to represent each *symmetric activity* sub-group. (3) To address the problem of clustering with asymmetric distance metric, we propose a *Seed-Representative-Centered clustering algorithm (SRC clustering)* to cluster people with asymmetric distance metric. We combine these contributions into a Group-Representative-based Activity Detection (GRAD) algorithm.

The rest of the paper is organized as follows. Section 2 describes the distance metric for modeling the activity correlation between two people, which is used in our SRC clustering. Section 3 describes the proposed *SAAS*. Section 4 describes the *SRC clustering algorithm*. Section 5 describes the definition of *group representative* and its use in the GRAD algorithm. Section 6 provides some discussions. Experimental results are shown in Section 7. We conclude the paper in Section 8.

(a) (b)

Fig. 1. Group activity example [8]. (a) Hierarchical Activity Structure. (b) The proposed *Symmetric-Asymmetric Activity* Structure.

II. THE ACTIVITY METRIC BETWEEN PEOPLE

Since the feature streams from different people may not be synchronized (e.g. inside a group, one person may act earlier than the other people), we use the Asynchronous Hidden Markov Model (AHMM) [1,7] to model the states between two people. AHMM was introduced to handle asynchronous feature streams. Using AHMM, the activity metric between person i and j under activity θ at time t can be calculated as:

$$co^{\theta}(i,j)=\sum_{k\in\theta}p(q_t=k\mid F_i^{1:t},F_j^{1:s}) \quad (1)$$

where, $F_i^{1:t}$ and $F_j^{1:s}$ are two asynchronous observation sequences for person *i* and *j*, $k\in\theta$ means all the states *q* that belong to the models of activity *θ*.

We call the *activity* between two people with the largest $co^{\theta}(i, j)$ *the label between i and j* ($L(i, j)$), which is defined in Eqn (2):

$$L(i,j)=\max_{\theta} co^{\theta}(i,j) \quad (2)$$

From Eqn (1) and (2), we can see that the activity metric is not symmetric ($co^{\theta}(i, j)$ and $L(i, j)$ may not equal to $co^{\theta}(j, i)$ and $L(j, i)$, respectively). Therefore, when we use this activity metric as the distance metric for clustering, we need to deal with the problem of clustering with asymmetric distance metric as will be described in detail in Section 5.

III. Symmetric and Asymmetric Activities

To solve the problem of hierarchical activity structure, we classify activities into *symmetric activities* and *asymmetric activities*. Assume we have two entities *A* and *B* (*A, B can be a single person or a subgroup of people*), the activity *θ* between *A* and *B* is defined as a *symmetric activity* if "*A* performing *θ* on *B*" is the same as "*B* performing *θ* on *A*". For example, the activity *WalkTogether* is a *symmetric activity* because "*A* and *B* are *walking together*" is the same as "*B* and *A* are *walking together*". From the above definition, we see that entities belonging to the same *symmetric activity* play similar roles for the activity and are interchangeable. We can further define the *symmetric group* as a group of entities where any two entities in the group perform the same *symmetric activity*. A *symmetric group* can have a flexible number of group members or entities. It should be noted that we also extend the definition of *symmetric group* to include single entity activity cases. For example, if a person walks alone and does not have any *symmetric activity* interaction with other people, this single person can form a *symmetric group of walking*.

Similarly, the activity *θ* between *A* and *B* is defined as an *asymmetric activity* if the activity is not a symmetric activity. For example, the activity *Following* is an *asymmetric activity* because "*A* is *following B*" is different from "*B* is *following A*".

With the introduction of *symmetric activity* and *asymmetric activity*, we proposed to solve the hierarchical activity structure problem by first clustering people into non-overlapping *symmetric group*s and then modeling the *asymmetric activity* interactions between the symmetric groups. We call this the *Symmetric-Asymmetric Activity Structure*. For example, in the example of Fig.1, we can first cluster people into two *symmetric groups*: the three-people fighting group (the dashed circle in Fig. 1(b)) and one person walking group (the dash-and-dotted circle in Fig. 1(b)). Then the asymmetric activity *approaching* between these four people can be modeled as the interaction between the fighting group and the walking group (the solid line circle in Fig. 1(b)). It should be noted that the idea of the proposed SAAS is general and can easily be extended to model other hierarchical activity structures. For example, we can model the *symmetric activities* of two *Walktogether* groups as the lower level activity and model the *symmetric activity Ignore* (i.e. people ignore each other) between these two groups as the higher level activity, thus form a *Symmetric-Symmetric Activity Structure* (SSAS).

IV. The SRC Clustering Algorithm

Based on the description of SAAS, before detecting the symmetric activity of each symmetric group and the asymmetric activity between symmetric groups, we need to cluster people into *symmetric groups* first. In this section, we propose an SRC clustering algorithm. The algorithm is described as follows:

(1) **Detecting the cluster seeds.** Two kinds of cluster seeds are defined.

a. Active people in the group. We define the *active* people as the people whose *change of body size* feature is larger than a threshold, which is defined as:

$$ob_i \text{ is an active person} \quad \text{if} \quad F^i_{Chang\_of\_Body\_Size} > 0.1 \quad (3)$$

$$\text{where } F^i_{Change\_of\_Body\_Size} = \left|SZ_i^t - SZ_i^{t-1}\right| \Big/ SZ_i^t, \quad SZ_i^t = W_i^t \cdot H_i^t$$

W_i^t *and* H_i^t *are the width and height of the Minimum Bounding Box of* ob_i

b. The people pair with high $co^{\theta}(i, j)$. People pairs with high $co^{\theta}(i, j)$ will also be considered as cluster seeds, if

$$\begin{cases} co^L(i,j) > 0.95 \text{ and } co^L(j,i) > 0.95, \\ L(i,j) = L(j,i), \text{ and} \\ L(i,j) \text{ is a symmetric activity} \end{cases} \quad (4)$$

where the definition of $co^L(i, j)$ and L(i,j) are the same as Eqn (1) and Eqn (2).

(2) **Post-processing of the cluster seeds.** After detecting the cluster seeds, a post processing process is performed to combine seeds that belong to the same *symmetric group*. Cluster seeds with the same symmetric activity label will be combined together. For example, if (*A*,*B*) is a cluster seed and *C* is another cluster seed, *C* can be combined with (*A*,*B*) to form a larger seed of (*A*,*B*,*C*) if L(*A*,*B*)=L(*A*,*C*)=L(*C*,*A*).

(3) **Calculate Seed Representatives (SR) for the cluster seeds.** We can combine people in the same cluster seed to create a *Seed Representative* for each cluster seed. In this paper, the average feature vector of people in the same seed is used as the SR for the cluster seeds.

(4) **Cluster the remaining people based on the SRs.** The calculated Seed Representatives serve as the center of each cluster and the rest people will be clustered around them. A person *K* will be grouped into the cluster indicated by the SR *A* if $co^L(A,K)$ is maximum and $L(A,K)$ is a *symmetric activity*. It should be noted that only the Seed-Representative-Centered (SR-Centered) metric value is used for clustering in this step. The SR-Centered metric value is defined as:

$$co^L(A,B) \text{ is an SR-Centered metric value}$$
$$\text{if} \quad A \text{ is a SR and } B \text{ is not a SR}$$

Since only the SR-Centered metric value is used for clustering, the asymmetry problem of the activity metric is avoided.

Since the SRC clustering algorithm extracts only high correlation pair in the seed detection step and use only SR-Centered value in the clustering step, it can deal with the problem of clustering with asymmetric distance metric.

V. Group Representative

As mentioned, people in the same *symmetric group* are interchangeable and play a similar role. Based on this property, each *symmetric group* can be represented by a single person, which we call the *Group Representative (GR)*. With the introduction of GR as well as our proposed SAAS and SRC clustering algorithm, we propose a Group-Representative-based Activity Detection (GRAD) algorithm to solve the problem of detecting group events with *varying number* of group members under the *hierarchical* activity structure. The GRAD algorithm can be summarized as follows:

(1) For each frame *t*, people are first clustered into non-overlapping *symmetric groups* by the SRC clustering algorithm (the dotted ellipses in Fig. 2). The *symmetric activity* for each symmetric group can then be recognized. We directly use the *activity label* for each cluster seed as the recognized activity for the symmetric group.

(2) Each *symmetric group* is represented by a Group Representative (the two bold solid circles in Fig. 2).

(3) The *asymmetric activity* between *symmetric groups* is then captured by the interaction of the GR of each *symmetric group* (the bold solid line in Fig. 2). As mentioned, the activity between two *symmetric groups* can also be symmetric (e.g. two groups *Ignore* each other). In this case, the interaction of the GR can also be used to detect the *symmetric activity* between two groups.

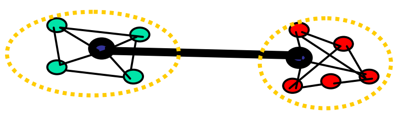

Fig. 2. Summary of the GRAD algorithm.

In the GRAD algorithm, since we use a single person (GR) to represent each *symmetric group*, we always have a fixed input feature vector length. Therefore, we can solve the problem of group event detection with varying group members.

There can be different ways to define the GR. In this paper, the GR is defined as the *most representative person* of the group which has the highest probability for the group's activity θ and also has the largest $co^{\theta}(i, j)$ value with other people in the *symmetric group*. Therefore, we define the GR as:

$$GR_{group\_A} = \max_i \left(p(F_i \mid \theta_{group\_A}) \cdot \prod_{\substack{j \in Group\ A \\ and\ j \neq i}} \phi^{\theta_{group\_A}}(ob_i, ob_j) \right) \qquad (5)$$

where F_i *is the feature vector of object i,* θ_{group_A} *is the activity for symmetric group A, and* $\phi^{\theta}(ob_i, ob_j) = \exp\left(co^{\theta}(ob_i, ob_j)\right)$.

In Eqn (3), $p(F_i|\theta_{group\_A})$ reflects the representativeness of person i for activity θ_{group_A}, and $\prod_{\substack{j \in Group\ A \\ and\ j \neq i}} \phi^{\theta_{group\_A}}(ob_i, ob_j)$ can be viewed as a prior which measures the distance of person *i* to other people in *symmetric group* A [11].

After the GR is detected for each *symmetric group*, the *asymmetric activity* between two *symmetric groups* can be detected based on the activity metric between GRs, as in Eqn (6).

$$\theta_{A,B} = \max_{\theta}\left(co^{\theta}(GR_A, GR_B) \cdot p_0(\theta)\right) \qquad (6)$$

where, $p_0(\theta) = \prod_{i \in A, j \in B} \phi^{\theta}(ob_i, ob_j)$ *is the prior for asymmetric activity* θ*, and* $\phi^{\theta}(ob_i, ob_j) = \exp\left(co^{\theta}(ob_i, ob_j)\right)$

A, B are two symmetric groups. $co^{\theta}(ob_i, ob_j)$ *is the same as Eqn (1).*

Since the activity metrics are not symmetric, we always put the GR whose group has smaller average speed in the first place of $co^{\theta}(GR_A, GR_B)$ (i.e. GR_A).

VI. Discussion

Since we have all the activity metrics between any two people, an alternative method to deal with the *detection-with-varying-number-of-members problem* is the Majority Vote (MV). (i.e. take the majority vote from all the asymmetric activity labels between *people pairs* from two symmetric groups as the resulting label). However, compared with MV, our proposed GR method has better results. The main reasons are: (a) When calculating the GR by Eqn (3), we are actually checking the whole symmetric group. The selected GR will have a global view of the whole group, and (b) When calculating the GR, we are also discarding the low-correlated *outlier people* from the asymmetric activity detection process, thus reducing the disturbance from these outlier people.

VII. Experimental Results

We use the BEHAVE dataset [8] and try to detect eight group activities: *InGroup, Approach, WalkTogether, Split, Ignore, Chase, Fight, RunTogether*. Example frames of the BEHAVE dataset is displayed in Fig. 1. The definitions of these eight activities are listed in Table 1. We classify these eight activities into two classes with *InGroup, WalkTogether, Ignore, Fight and RunTogether* as *symmetric activities*, and *Approach, Split and Chase* as asymmetric activities. It should be noted that we extended the definition of activity *Ignore*. The two people will *ignore* each other if they do not have other activity correlation. Furthermore, *Ignore* will also be used to model the non-interaction case between two symmetric groups. We also add a *single* activity into the *symmetric activity* list for those people that cannot be clustered into any *symmetric group.*

Six features are used for calculating the persons' activity metrics in Eqn (1). All the features are derived from the persons' ground-truth Minimum Bounding Box (MBB) information which is available in the BEHAVE dataset. They are (1) *Change of MBB Width,* (2) *Change of MBB Height*, (3) *Avg_Speed,* (4) *Distance to the mean,* (5) *Speed variance, and* (6) *Motion Direction.* (Note: The definition of *Change of MBB Width, Change of MBB Height,* and *Avg_Speed* are the same as in [9], *Distance to the mean* is the distance from the individual to the center of the *people pair*, Speed variance is the speed difference between each individual and the mean speed of the *people pair*, and *Motion Direction* is the angle between the individual's motion direction and the line linking the *people pair*).

Table 1 The definition of group activities

Activity	Definition
InGroup	The people are in a group and not moving very much
Approach	Two people or groups with one (or both) approaching the other
WalkTogether	People walking together
Split	Two or more people splitting from one another
Ignore	Ignoring of one another
Chase	One group chasing another
Fight	Two or more groups fighting
RunTogether	The group is running together

When calculating the persons' activity metrics by Eqn (3), we use two hidden states for each activity. The emission probability of each hidden state is modeled by a Gaussian Mixture Model (GMM) with two Gaussian mixtures.

We separated the labeled part of the dataset into 6 sequences. In our experiment, we randomly select three sequences for training and three sequences for testing. Five independent experiments are performed and the error rates are averaged. The experimental results are shown in Table 2. In Table 2, two methods are compared:

(I) SAAS+SRC+MV Based on the proposed SAAS, use our proposed SRC clustering algorithm to cluster people into *symmetric groups* and detect the activity of these *symmetric groups*, then use the Majority Vote to detect the asymmetric activities between the symmetric groups.

(II) The GRAD algorithm (SAAS+SAC+GR). Use the GRAD algorithm to detect group activities.

It should be noted that both of these two methods use our proposed SAAS and SRC clustering to cluster people and detect symmetric activities. However, Method I uses MV to detect asymmetric activities and Method II uses GR to detect asymmetric activities.

In Table 2, two error rates are computed: the Event Detection Error Rate (EDER) and the Group Clustering Error Rate (GCER), they are defined in Eqn (7) and (8) respectively.

$$EDER = \# \text{ of error frames} \Big/ \# \text{ of total frames} \qquad (7)$$

where t is an error frame if any of the following take place at t

(1) any object in t is misclustered into another symmetric group

(2) any of the symmetric activities is misclassified

(3) any of the asymmetric activities is misclassified

$$GCER = \# \text{ of clustering error frames} \Big/ \# \text{ of total frames} \qquad (8)$$

where t is an clustering error frame if the following take place at t

any object in t is misclusted into another symmetric group

The *EDER* reflects the overall performance of the algorithm in detecting both the *symmetric activities* and the *asymmetric activities*. And the GCER reflects the performance of the algorithm in clustering people into *symmetric groups*.

Several observations from Table 2 are listed below:

(1) Since both methods use the proposed SRC clustering algorithm for clustering people into *symmetric groups*, their GCERs are the same. The low GCER demonstrates the effectiveness of the SRC clustering algorithm.

(2) Comparing the EDER, we can see that the proposed GRAD algorithm has better EDER than that uses majority vote. This supports our claim that the introduction of GR can greatly improve the detection rate for asymmetric activities.

(3) The EDER of the GRAD algorithm is close to the GCER. This implies the fact that most of the errors come from the mis-clustering of people. The performance of the GRAD algorithm can be further improved if people can be clustered more correctly into symmetric groups.

(4) In our experiment, we use 2 hidden states for each activity and 2 Gaussians for each state. The performance may be further improved if we use more hidden states and Gaussians.

(5) Besides GR, our proposed SAAS and SRC clustering algorithms can also handle group event detection with *varying group members* under *hierarchical activity structures*. This is another major contribution of the paper which is not reflected in Table 2.

Table 2 The experimental results for the GRAD algorithm

	SAAS+SRC+MV	SAAS+SRC+GR (GRAD)
GCER	7.4%	7.4%
EDER	18.6%	10.2%

Table 3 shows the average False Alarm rate (FA) and Miss Detection rate (Miss) [9] of the GRAD algorithm for the activities in Table 1.

Table 3 The average Frame Level FA and Miss for GRAD

Activity		GRAD	Activity		GRAD
Ingroup	Miss (%)	1.2	Ignore	Miss (%)	5.5
	FA (%)	2.08		FA (%)	6.76
RunTogether	Miss (%)	17.4	Approach	Miss (%)	8.9
	FA (%)	0.22		FA (%)	3.87
WalkTogether	Miss (%)	10.1	Splict	Miss (%)	10.6
	FA (%)	3.85		FA (%)	1.11
Fight	Miss (%)	24.7	Chase	Miss (%)	30.8
	FA (%)	0.82		FA (%)	0.72

VIII. CONCLUSION

In this paper, we proposed (1) a *symmetric-asymmetric activity structure* for the detection with hierarchical activity structure, (2) a *Group Representative* to handle the group event detection with varying number of group members, and (3) an *SRC clustering algorithm* to deal with clustering with asymmetric distance metric. Experimental results demonstrate the effectiveness of our proposed algorithm.

ACKNOWLEDGMENT

This work was supported in part by the following grants: ARO PECASE Grant (W911NF-05-1-0491) and ARO MURI Grant (W 911 NF 0710287). The authors would like to thank Dr. Samy Dengio for providing part of the code for implementing the AHMM.

REFERENCES

[1] D. Zhang, D. Gatica-Perez, S. Bengio and I. McCowan, "Modeling individual and group actions in meetings with layered HMMs," *IEEE Trans. Multimedia*, vol. 8, pp. 509-520, 2006.

[2] N. Oliver, E. Horvitz, and A. Garg. "Layered representations for learning and inferring office activity from multiple sensory channels," *Proc. ICMI*, October 2002.

[3] S. Park and J.K. Aggarwal, "A hierarchical bayesian network for event recognition of human actions and interactions," *Association for Computing Machinery Multimedia Systems Journal*, 2004.

[4] S. Hongeng and R. Nevatia, "Multi-agent event recognition," in *Proc. IEEE Int'l. Conf. Computer Vision*, July 2001.

[5] N. Vaswani, A.R. Chowdhury and R. Chellappa, "Activity recognition using the dynamic of the configurations of interacting objects," *IEEE Conf. Computer Vision and Pattern Recognition*, 2003.

[6] D. Wyatt, T. Choudhury and J. Bilmes, "Conversation detection and speaker segmentation in privacy-sensitive situated speech data," *Speech and audio processing for intelligent environments*, 2007.

[7] S. Bengio, "An asynchronous hidden Markov model for audio-visual speech recognition," S. *Proc. NIPS 15*, 2003.

[8] BEHAVE data, http://groups.inf.ed.ac.uk/vision/behavedata/interactions/.

[9] W. Lin, M.-T. Sun, R. Poovendran and Z. Zhang, "Activity Recognition using a Combination of Category Components and Local Models for Video Surveillance," *IEEE Trans. CSVT*, no. 8, 2008.

[10] H. Späth, "Cluster Analysis Algorithms for Data Reduction and Classification of Objects," *Halsted Press*, 1980.

[11] K. Smith, D. Gatica-Perez and J.M. Odobez, "Using Particla Filters to Track Varying Number of Interacting People," CVPR, 2005.