

Group Event Detection with a Varying Number of Group Members for Video Surveillance

Weiyao Lin, Ming-Ting Sun, *Fellow, IEEE*, Radha Poovendran, *Senior Member, IEEE*,
and Zhengyou Zhang, *Fellow, IEEE*

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 a varying number of group members, and use an asynchronous hidden Markov model (AHMM) to model the relationship between people. Furthermore, we propose a group activity detection algorithm which can handle both 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.

Index Terms—Group event detection, group representative, varying group member, video surveillance.

I. INTRODUCTION

DETECTING human group behavior or human interactions has attracted increasing research interests [1]–[6]. Some examples of group events include people fighting, people being followed, people walking together, terrorists launching attacks in groups, etc. Being able to automatically detect group activities of interest is important for many security applications. In this paper, we address the following issues for group event detection.

A. Group Event Detection with a Varying Number of Group Members

Most previous group event detection research [1], [2] uses a hidden Markov model (HMM) or its variation to model the human interactions. Some researchers try to recognize human interactions based on a content-independent semantic

Manuscript received January 28, 2009; revised July 16, 2009. Date of publication July 26, 2010; date of current version August 4, 2010. This work was supported in part by the ARO PECASE, under Grant W911NF-05-1-0491, in part by the ARO MURI, under Grant W 911 NF 0710287, in part by the Chinese National 973, under Grant 2010CB731401 and Grant 2010CB731406, and in part by the National Science Foundation of China, under Grant 60632040, Grant 60928003, and Grant 60973067. This paper was recommended by Associate Editor R. Green.

W. Lin is with the Institute of Image Communication and Information Processing, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: wylin@sjtu.edu.cn).

M.-T. Sun and R. Poovendran are with the Department of Electrical Engineering, University of Washington, Seattle, WA 98195 USA (e-mail: mts@u.washington.edu; rp3@u.washington.edu).

Zhengyou Zhang is with Microsoft Research, Microsoft Corporation, Redmond, WA 98052-8300 USA (e-mail: zhang@microsoft.com).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TCSVT.2010.2057013

set [3], [4]. However, most of these works are designed to recognize group activities with a fixed number of group members, where the dimension of the input feature vectors is fixed. They cannot handle the case where the number of group members is varying, which often occurs in our daily life (e.g., people may leave or join a group activity). In this case, the input feature vector length varies with the number of group members. Although some previous works [5], [6] tried to deal with the detection of group activities with a varying number of members, most of them have assumptions under some specific scenarios which 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 of each other and they may further interact among them to form a hierarchical structure. For example, in Fig. 1, 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). 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. Our work addresses 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 i following person j ” is not the same as “person j following person i ”). 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).



Fig. 1. Example of a group event with hierarchical activity structure [8].

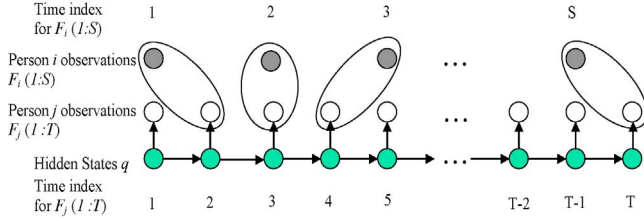


Fig. 2. AHMM for the observation of independent individuals i and j over the time periods $1:S$ and $1:T$, respectively.

- 2) To address the problem of detecting events with a varying number of people, we propose to use a group representative (GR) to represent each symmetric activity subgroup.
- 3) To address the problem of clustering with an asymmetric distance metric, we propose a seed-representative-centered (SRC) clustering algorithm to cluster people with asymmetric distance metric.
- 4) To combine these contributions into a group-representative-based activity detection (GRAD) algorithm.

The rest of the paper is organized as follows. Section II describes the distance metric for modeling the activity correlation between two people, which is used in our SRC clustering. Section III describes the proposed SAAS. Section IV describes the SRC clustering algorithm. Section V describes the definition of group representative and its use in the GRAD algorithm. Experimental results are shown in Section VI. Section VII discusses some possible extensions of the algorithm. We conclude the paper in Section VIII.

II. ACTIVITY CORRELATION METRIC BETWEEN PEOPLE

In this paper, we use the asynchronous hidden Markov model (AHMM) [1], [7] to model the activity correlation metric between two people. It should be noted that our proposed GRAD algorithm, as to be detailed later, is general and can be easily extended to use other models [2], [12]–[14].

AHMM was introduced to handle asynchronous feature streams. As in Fig. 2, assume there are two asynchronous observation (or feature) sequences $F_i(1:S)$ for person i from time 1 until time S and $F_j(1:T)$ for person j from time 1 until time T with the length $T \geq S$, the AHMM tries to associate the corresponding features in order to obtain a better match between streams.

The probability that the system emits the next observation of sequence F_i at time t while in state $q(t) = k$, as defined in

[7], is

$$\varepsilon(k, t) = P(\tau(t) = s \mid \tau(t-1) = s-1, q(t) = k, F_i(1:s), F_j(1:t)) \quad (1)$$

where $P(\cdot)$ represents the probability. The additional hidden variable $\tau(t) = s$ can be seen as the alignment between F_i and q (and F_j which is also aligned with q). Based on (1), we can define the forward procedure as in (2) [7]

$$\alpha(s, k, t) = p(\tau(t) = s, q(t) = k, F_i(1:s), F_j(1:t)) \quad (2)$$

$$\begin{aligned} &= \varepsilon(k, t) p(F_i(s), F_j(t) \mid q(t) = k) \sum_{l=1}^N P(q(t) = k \mid q(t-1) = l) \\ &\quad \alpha(s-1, l, t-1) + (1 - \varepsilon(k, t)) p(F_j(t) \mid q(t) = k) \\ &\quad \sum_{l=1}^N P(q(t) = k \mid q(t-1) = l) \alpha(s, l, t-1) \end{aligned}$$

where $p(\cdot)$ represents the distribution, $F_i(s)$ and $F_j(t)$ are the observations for persons i and j at time s and t , respectively, and N is the total number of hidden states.

Therefore, the activity correlation metric for person j with respect to person i under activity θ at time t can be calculated as

$$\begin{aligned} co_i^\theta(j, t) &= \sum_{k \in \theta} P(q(t) = k \mid F_i(1:s), F_j(1:t)) \\ &= \frac{\sum_{k \in \theta} \sum_{s=t-\Delta t}^{s=t+\Delta t} \alpha(s, k, t)}{\sum_{k=1}^{N_s} \sum_{s=t-\Delta t}^{s=t+\Delta t} \alpha(s, k, t)} \quad (3) \end{aligned}$$

where $k \in \theta$ means all the states that belong to the model of activity θ . N_s is the total number of states over all activities.

We call the activity with the largest $co_i^\theta(j, t)$ the label for j with respect to i , ($L_i(j)$), which is defined in (4)

$$L_i(j) = \max_{\theta} co_i^\theta(j, t). \quad (4)$$

The reason of using AHMM for modeling the activity correlation metric is that AHMM can handle asynchronous feature streams. Since the feature streams of different people in the same group may not be perfectly synchronized (e.g., when two people walk together, one person may stretch the leg earlier than the other person), AHMM can help reduce the possible recognition errors from these action asynchronies, as will be demonstrated in the experimental results.

Also, from (3) and (4), we can see that the activity correlation metric is not symmetric (e.g., $co_i^\theta(j, t) = \sum_{k \in \theta} P(q(t) = k \mid F_i(1:s), F_j(1:t)) \neq co_j^\theta(i, t) = \sum_{k \in \theta} P(q(t) = k \mid F_j(1:s), F_i(1:t))$ because the order of F_i and F_j has been changed, and similarly $L_i(j)$ may not be equal to $L_j(i)$). Therefore, when we use this activity correlation metric as the distance metric for clustering, we need to deal with the problem of clustering with an asymmetric distance metric as will be described in detail in Section IV.

III. SYMMETRIC AND ASYMMETRIC ACTIVITIES

To solve the problem of the hierarchical activity structure, we classify activities into symmetric activities and asymmetric activities. Assume we have two entities i and j , the activity θ between i and j is defined as a symmetric activity if “ i has the activity θ with j ” is the same as “ j has the activity θ with i .” For example, the activity *WalkTogether* is a symmetric activity because “ i walking together with j ” is the same as “ j walking together with i .” 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 varying 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 i and j is defined as an asymmetric activity if the activity is not a symmetric activity. For example, the activity *Following* is an asymmetric activity because “ i is following j ” is different from “ j is following i .”

With the introduction of symmetric activity and asymmetric activity, we proposed to solve the hierarchical-activity-recognition problem by first clustering people into non-overlapping symmetric groups and then modeling the asymmetric-activity interactions between the symmetric groups. We call this the SAAS. For example, in the example of Fig. 1, we can first cluster people into two symmetric groups: the three-people fighting group and one person walking group. Then the asymmetric activity *Approaching* between these four people can be modeled as the interaction between the fighting group and the walking group. It should be noted that the idea of the proposed SAAS is general and can be easily 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 forms a symmetric-symmetric activity structure (SSAS).

IV. 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 a SRC clustering (SRC clustering) algorithm. The algorithm is described as follows.

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

a) *Active people in the group*. Person i will be considered as an *active* person in the group if

$$C_i(t) > T_C \quad (5)$$

where $C_i(t)$ is the change of body size of person i at time t and T_C is a threshold. $C_i(t)$ is calculated

by

$$C_i(t) = \frac{|W_i(t) \cdot H_i(t) - W_i(t-1) \cdot H_i(t-1)|}{(W_i(t) \cdot H_i(t))}$$

where $W_i(t)$ and $H_i(t)$ are the width and height of the minimum bounding box (MBB) (which is the smallest rectangular box that includes the person in motion [9]) of person i at time t .

b) *The people pairs with high activity correlation metric values*. People pair i and j with high activity correlation metric values will also be considered as cluster seeds if

$$\begin{cases} co_i^L(j, t) > T_o \text{ and } co_j^L(i, t) > T_o \\ L_i(j) = L_j(i), \text{ and} \\ L_i(j) \text{ is a symmetric activity} \end{cases} \quad (6)$$

where T_o is a threshold to decide where people pairs i and j have high activity correlation.

Step 2) Post-processing of the cluster seeds. After detecting the cluster seeds, a post-processing 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 $L_a(b) = L_a(c) = L_c(a)$.

Step 3) Calculate seed representatives (SRs) for the cluster seeds. We can combine people in the same cluster seed to create a SR for each cluster seed. There can be many ways to define the SR. For example, we could pick any feature vector close to the cluster center as the SR. In this paper, the average feature vector of people in the same seed is used as the SR for the cluster seeds.

Step 4) Cluster the remaining people based on the SRs. The calculated SRs serve as the centers of clusters and the remaining people are clustered around them. A person i is grouped into the cluster indicated by the SR K if $co_i^L(K, t)$ is maximum and $L_i(K)$ is a symmetric activity. It should be noted that only the SRC metric value is used for clustering in this step. The SRC metric value is defined as $co_i^L(K, t)$ is an SRC metric value if K is a SR and i is not a SR.

Since only the SRC metric value is used for clustering, the asymmetry problem of the activity correlation metric is avoided.

As a summary, the proposed SRC clustering algorithm extracts only high correlation pairs as well as single active person in the seed detection step and use only the SRC value in the clustering step. Therefore, it can deal with the problem of clustering with an asymmetric distance metric.

V. GROUP REPRESENTATIVE AND THE GRAD ALGORITHM

A. Definition of 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 entity, which we call the group representative (GR). There can be different ways to define the GR. In this paper, we investigate three ways to define the GR. They are described as follows.

1) *Physical GR (P-GR)*: The Physical GR is an actual person selected from the symmetric group. We define P-GR as the most representative person of the symmetric group which has the highest joint value for representing the group's activity θ_A as well as correlating with other people in the symmetric group, as in (7)

$$P-GR_A(t) = \max_i (p(F_i(t)|\theta_A) \cdot p_0(i, \theta_A, t)) \quad (7)$$

where $P-GR_A(t)$ is the P-GR for symmetric group A at time t , $F_i(t)$ is the feature vector of person i at time t , θ_A is the activity

for A , and $p_0(i, \theta_A, t) = \exp\left(\sum_{\substack{j \in A \\ \text{and } j \neq i}} co_j^{\theta_A}(i, t)\right)$. In

(7), $P(F_i(t)|\theta_A)$ reflects the representativeness of person i for activity θ_A , and $p_0(i, \theta_A, t)$ can be viewed as a prior which measures the distance or correlation of person i to other people in A [11].

2) *Virtual GR (V-GR)*: The virtual GR is not an actual person. Rather, it is the combination of multiple people in the same symmetric group. The V-GR is defined as the average of *all* people in the feature space in the same symmetric group. Therefore, the feature vector of V-GR at time t can be defined as

$$F_{V-GR_A}(t) = \text{avg}_{i \in A}(F_i(t)) \quad (8)$$

where $F_i(t)$ is the feature vector for person i at time t , and group A is the symmetric group.

3) *Selective Virtual GR (SV-GR)*: Similar to V-GR, SV-GR is also a virtual GR which is the combination of multiple people. However, SV-GR is the average of only those most representative people for the symmetric group, as in (9)

$$F_{SV-GR_A}(t) = \text{avg}_{i \in R_A}(F_i(t)) \quad (9)$$

where $F_{SV-GR_A}(t)$ is the feature vector of SV-GR for group A at time t , $F_i(t)$ is the feature vector for person i at time t . $R_A = \{i | N(p(F_i(t)|\theta_A) \cdot p_0(i, \theta_A, t)) > T_R\}$, where T_R is a threshold to decide whether person i is representative. $N(\cdot)$ is the normalization operation such that $\sum_i N(p(F_i(t)|\theta_A) \cdot p_0(i, \theta_A, t)) = 1$.

B. GRAD Algorithm

With the introduction of GR as well as our proposed SAAS and SRC clustering algorithm, we propose a GRAD algorithm to solve the problem of detecting group events with a varying number of group members under a hierarchical activity structure. The GRAD algorithm can be summarized as follows.

Step 1) For each frame t , people are first clustered into non-overlapping symmetric groups by the SRC clustering algorithm (the dotted ellipses in Fig. 3). The symmetric activity for each symmetric group can then be recognized. In this paper, we propose the following two methods to recognize the symmetric activity.

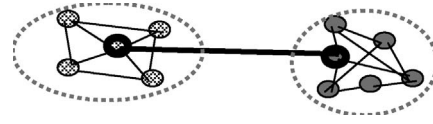


Fig. 3. GRAD algorithm.

- a) Directly use the activity label for each cluster seed as the recognized activity for the symmetric group.
- b) A more sophisticated way is to extract some group features [5, 15] from the symmetric group and use a separate model such as HMM for recognition, as described by (10)

$$\theta_A(t) = \max_{\theta} (p(F_A(t)|\theta) \cdot p_1(\theta, t)) \quad (10)$$

where $p_1(\theta, t) = \exp\left(\sum_{i, j \in A} co_i^{\theta}(j, t)\right)$ can be viewed as a prior for activity [11]. $F_A(t)$ is the global feature vector for symmetric group A , and $P(F_A(t)|\theta)$ is the probability calculated by the model used for recognizing symmetric activities.

Step 2) Each symmetric group is represented by a GR (the two bold solid circles in Fig. 3).

Step 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. 3). In this paper, we detect the asymmetric activity between two symmetric groups based on the activity correlation metric between GRs, as in (11)

$$\theta_{A,B}(t) = \max_{\theta} (co_{GR_B}^{\theta}(GR_A, t) \cdot p_2(\theta, t)) \quad (11)$$

where $p_2(\theta, t) = \exp\left(\sum_{i \in A, j \in B} co_j^{\theta}(i, t)\right)$ is the prior for asymmetric activity θ . A and B are two symmetric groups. Since the activity correlation metrics are not symmetric, in our notations, we put the GRs in the order according to a specific feature such as the average speed of the symmetric group (i.e., the average speed for group A is smaller than B in $co_{GR_B}^{\theta}(GR_A)$).

Furthermore, 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.

In the GRAD algorithm described above, since we use a single 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 a varying number of group members.

C. Discussion of GR

Since we have all the activity correlation metrics between any two people, there can be alternative methods to deal with the detection-with-a-varying-number-of-members problem. For example, we can use the majority vote (MV) method [17], [18] for asymmetric activity recognition by taking the MV from all the asymmetric activity labels between people

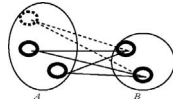


Fig. 4. Example of the disturbance from an outlier person (dotted circle: outlier person; bold-faced circle: regular person).

pairs from two symmetric groups as the resulting activity label. Compared with MV and other methods, the major difference of our proposed GR method is to use a single representative (physical or virtual) to represent the whole symmetric group. With the introduction of GR, we can have the following advantages.

1) Methods such as MV lack a global view of the whole group since all the activity correlation metrics only reflect the local information between two people. On the other hand, when selecting the GR by (7)–(9), we are actually checking the whole symmetric group. Therefore, the selected GR will have a global view of the whole group.

2) More importantly, when detecting the asymmetric activity between two symmetric groups, some people that are not highly related may disturb the recognition result. For example, as in Fig. 4, the asymmetric activity θ between A and B is mainly decided by the interaction between the bold-faced people (i.e., bold-faced circles in Fig. 4) in A and the bold-faced people in B. The dotted person located on the side of A does not have high correlation in θ with people in B and may have misclassified activity label with B. This dotted person is an outlier and may disturb the recognition results. When using methods such as MV to perform recognition, the dotted outlier person is included and the recognition accuracy may be decreased. However, when using GR with our proposed method (especially the P-GR and the SV-GR), the low-correlated outlier person will be discarded from the asymmetric activity detection process, thus reducing the disturbance from these outlier people. Therefore, our proposed GR will also increase the recognition accuracy by efficiently discarding outliers.

VI. EXPERIMENTAL RESULTS

In this section, we show experimental results for our proposed methods and compare our results with other methods. We perform experiments based on the BEHAVE dataset [8]. Six long sequences are selected in our experiments with each sequence including 7000 to 11000 frames. We try to detect eight group activities: *InGroup*, *Approach*, *WalkTogether*, *Split*, *Ignore*, *Chase*, *Fight*, and *RunTogether*. Some example video frames are shown in Fig. 5. The definitions of these eight activities are listed in Table I. We classify these eight activities into two categories 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



Fig. 5. Some example video frames for the group activities [8]. (a) *InGroup*. (b) *Fight*. (c) *WalkTogether*.

TABLE I
DEFINITION OF GROUP ACTIVITIES (ACTIVITIES IN GRAY ARE SYMMETRIC ACTIVITIES AND ACTIVITIES IN WHITE ARE ASYMMETRIC ACTIVITIES)

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

TABLE II
DEFINITION OF INPUT FEATURES

Feature Name	Definition
Change of Width	$\left \frac{W_i(t) - W_i(t-1)}{W_i(t)} \right $
Change of Height	$\left \frac{H_i(t) - H_i(t-1)}{H_i(t)} \right $
Speed	$\sqrt{(x_i(t) - x_i(t-1))^2 + (y_i(t) - y_i(t-1))^2}$
Average Distance	$\sqrt{\left(x_i(t) - \frac{x_i(t) + x_j(t)}{2}\right)^2 + \left(y_i(t) - \frac{y_i(t) + y_j(t)}{2}\right)^2}$
Speed Difference	$\frac{\left(\sqrt{(x_i(t) - x_i(t-1))^2 + (y_i(t) - y_i(t-1))^2} - \sqrt{(x_j(t) - x_j(t-1))^2 + (y_j(t) - y_j(t-1))^2} \right)}{2}$
Motion Direction Angle	$\arctan\left(\frac{y_i(t) - y_i(t-1)}{x_i(t) - x_i(t-1)}\right) - \arctan\left(\frac{y_j(t) - y_j(t)}{x_j(t) - x_i(t)}\right)$
<i>Note:</i> The features in this table forms an input feature vector for i when calculating its correlation with j . $[x_i(t), y_i(t)]$ is the center of MBB for i at time t . $W_i(t)$ and $H_i(t)$ is the width and height of the MBB for i at time t , respectively.	

For simplicity, we only use the MBB information [9] to derive all the features used for group activity recognition. Note that the proposed algorithm is not limited to the MBB features. Other more sophisticated features [19], [20] can easily be applied to our algorithm to give better results. Six features are used to calculate the persons' activity correlation metrics in (3). They are listed in Table II.

In order to exclude the effect of the tracking algorithm, we use the ground-truth tracking data which is available in the

TABLE III
DEFINITION OF GROUP FEATURES

Feature Name	Definition
Average Change of Width	$\sum_{i \in A} \left(\frac{ W_i(t) - W_i(t-1) }{W_i(t)} \right) / \sum_{i \in A} 1$
Average Change of Height	$\sum_{i \in A} \left(\frac{ H_i(t) - H_i(t-1) }{H_i(t)} \right) / \sum_{i \in A} 1$
Average Speed	$\sum_{i \in A} \sqrt{(x_i(t) - x_i(t-1))^2 + (y_i(t) - y_i(t-1))^2} / \sum_{i \in A} 1$
Average Distance	$\sum_{i \in A} \left(\sqrt{\left(x_i(t) - \frac{\sum_{j \in A} x_j(t)}{\sum_{j \in A} 1} \right)^2 + \left(y_i(t) - \frac{\sum_{j \in A} y_j(t)}{\sum_{j \in A} 1} \right)^2} \right) / \sum_{i \in A} 1$
Speed variance	$\sum_{i \in A} \left(\sqrt{(x_i(t) - x_i(t-1))^2 + (y_i(t) - y_i(t-1))^2} - avg_speed \right)^2 / \sum_{i \in A} 1$
Note: the definition of $x_i(t)$, $y_i(t)$, $W_i(t)$ and $H_i(t)$ are the same as in Table II. A is a symmetric group.	

BEHAVE dataset to get the MBB information. In practice, various practical tracking methods [15], [21] can be used to obtain the MBB information. Furthermore, the thresholds T_c , T_o , and T_R in (5), (6), and (9) are set to be 0.1, 0.95, and 0.3, respectively. These values are manually selected based on the statistics from one of the training sets. In practice, these thresholds can also be selected by some more sophisticated ways such as the validation set method [9].

In our experiments, four methods are compared. For all the HMMs or AHMMs in these methods, we use two hidden states for each activity (plus the starting state and the finishing state, there are in total four states) and a two-mixture Gaussian mixture model [23], [24] for modeling the emission probability for each hidden state. It should be noted that the methods selected to compare in our experiments are typical and the results can easily be extended to other related methods [2], [13], [16], [19]. The four methods are described as follows.

1) *HMM*: Use a single HMM [12], [21] to recognize either the symmetric activities or the asymmetric activities. When recognizing symmetric activities, the group features in Table III are calculated for each symmetric group. However, it should be noted that the traditional HMM cannot deal with the recognition of hierarchical-structure activities (i.e., a single HMM cannot recognize a lower level symmetric activity and an upper-level asymmetric activity at the same time). Furthermore, since the input feature vector length is fixed, it also cannot recognize an activity with a varying number of group members.

2) *Layered HMM + SAAS*: In [1], a layered HMM was proposed. In our experiment, we extend this layered HMM based on our proposed SAAS to recognize hierarchical-structure group activities, where the HMMs in the lower layer recognize the symmetric activities for each symmetric subgroup and the HMM in the higher layer takes the outputs of the lower

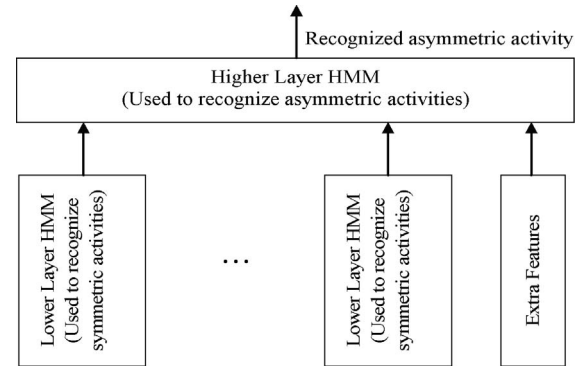


Fig. 6. Layered HMM (lower layer HMMs are used to recognize symmetric activities and a higher layer HMM is used to recognize asymmetric activities).

layer as input to recognize asymmetric activities, as in Fig. 6. Furthermore, extra features are also calculated as input to the higher layer HMM [1]. In our experiment, we use hard decision outputs [1] of the lower layer HMMs as the input to the higher layer HMM. Furthermore, features in Table II are used as the extra features for inputting to the higher layer HMM. The extra features are calculated between two symmetric subgroups. However, similar to HMM, since the input feature vector length of the layered HMM is also fixed, it cannot deal with the problem of activity recognition with a varying number of group members.

3) *SAAS + SRC + MV*: Based on the proposed SAAS, it uses our proposed SRC clustering algorithm to cluster people into symmetric groups and detect the activity of these symmetric groups, then uses the MV to detect the asymmetric activities between the symmetric groups. When detecting the symmetric activities, two different methods are used: 1) use the activity label for each cluster seed as the recognized

TABLE IV
CAPABILITIES OF DIFFERENT METHODS IN DEALING WITH DIFFERENT
EXPERIMENTAL TASKS

	HMM	Layered HMM	SAAS + SRC +MV	GRAD (SAAS+ SRC + GR)
Symmetric + Fixed	○	○	○	○
Asymmetric + Fixed	○	○	○	○
Hierarchical + Fixed	×	○	○	○
Hierarchical + Varying	×	×	○	○

Note: Label “○” means the method is able to deal with the corresponding task, the label “×” means the method is unable to deal with the corresponding task.

activity for the symmetric group (SAAS+SRC+MV-1 in Tables V–VII), and 2) calculate the group features from the symmetric group and use the HMM model for recognition, as in (10) (SAAS+SRC+MV-2 in Tables V–VII). The SAAS+SRC+MV method can recognize hierarchical-structure activities as well as activities with a varying number of group members. However, it should be noted that using only MV cannot recognize hierarchical-structure activities and varying-member activities. By combining MV with our proposed SAAS and SRC clustering algorithm, it can deal with these activities.

4) *GRAD Algorithm (SAAS + SRC + GR)*: Use our proposed SAAS and SRC clustering to cluster people and detect symmetric activities. However, different from the SAAS+SRC+MV method which uses MV to detect asymmetric activities, the GRAD algorithm uses our proposed GR to detect asymmetric activities. Similar to the SAAS+SRC+MV method, we use two different methods to detect symmetric activities. They are: 1) use the cluster seed label as the recognized activity (GRAD-1 in Tables V–VII), and 2) use an independent HMM to recognize the symmetric activities (GRAD-2 in Tables V–VII).

Experiments for four scenarios are designed to compare the four methods described above, they are: 1) recognizing *only symmetric activities* with a *fixed number* of group members (*symmetric + fixed* in Table IV); 2) recognizing *only asymmetric activities* with a *fixed number* of group members (*asymmetric + fixed* in Table IV); 3) recognizing *hierarchical-structure activities* with a *fixed number* of group members (*hierarchical + fixed* in Table IV); and 4) recognizing *hierarchical-structure activities* with a *varying number* of group members (*hierarchical + varying* in Table IV). These four sets of experiments will be described in detail in the following sections. Table IV summarizes the capabilities of the four methods in dealing with these four experimental tasks. It should be noted that the scenario of *hierarchical + varying* is the general case for group activities and the other scenarios can be viewed as the special cases for this scenario.

A. Experimental Results for Recognizing Only Symmetric Activities with a Fixed Number of Group Members

In this section, we compare the performances of the four methods in recognizing only the four symmetric activities (i.e., *InGroup*, *WalkTogether*, *Fight*, and *RunTogether*). Furthermore, we assume that the symmetric groups have already been clustered and the number of members in all symmetric

TABLE V
TFER COMPARISON FOR SYMMETRIC ACTIVITY RECOGNITION WITH
FIXED NUMBER OF GROUP MEMBERS

Methods	TFER
Set-1 (HMM, Layered HMM+SAAS, SAAS+SRC+MV-2, and GRAD-2)	5.36%
Set-2 (SAAS+SRC+MV-1 and GRAD-1)	5.52%

groups is fixed to 3. In order to fix the member for all groups to 3, we discard the activity segments from the dataset whose group members are less than 3. For activity segments with more than three members, we manually pick three members to form a symmetric group.

We perform experiments under 50% training and 50% testing. Five independent experiments are performed and the results are averaged. The experimental results are listed in Table V. In Table V, the total frame error rate (TFER) [9, 25] is compared. TFER is defined by N_{t_miss}/N_{t_f} , where N_{t_miss} is the total number of misdetection frames for all activities, and N_{t_f} is the total number of frames in the test set. TFER reflects the overall performance of each algorithm in recognizing all these five symmetric activities.

Since only symmetric activities with a fixed number of people are recognized in this experiment, the HMM method, the Layered HMM+SAAS method, the SAAS+SRC+MV-2 method, and the GRAD-2 method are exactly the same to each other and they can be classified as one set (Set-1 in Table V). Similarly, the SAAS+SRC+MV-1 method and the GRAD-1 method can be classified as another set (Set-2 in Table V). Basically, the major difference between the methods of these two sets is that methods in Set-1 can have a global view of the whole symmetric group by using the group features, while the methods in Set-2 only use local information of the cluster seeds for recognition. However, from Table V, we can see that the TFER for both sets are very close. Similar results can also be found for larger numbers of group members. This implies that since members in the symmetric group are interchangeable and similar, using only local information from parts of the group members may be enough to recognize symmetric activities.

B. Experimental Results for Recognizing Only Asymmetric Activities with a Fixed Number of Group Members

In this section, we perform experiments to recognize the three asymmetric activities (*Approach*, *Split*, and *Chase*). Similar to the previous section, we fixed the number of members in each asymmetric group to 4. We also assume that each asymmetric group contains two symmetric subgroups with one group containing three people and the other group containing one person. It should be noted that since the number of group member is fixed in this experiment, the SRC clustering is not needed for the SAAS+SRC+MV method and the GRAD method and thus is skipped.

The TFER result comparison for asymmetric activity recognition under 50% training and 50% testing is shown in Table VI.

TABLE VI

TFER COMPARISON FOR ASYMMETRIC ACTIVITY RECOGNITION WITH FIXED NUMBER OF GROUP MEMBERS

Methods	TFER
HMM	23.36%
Layered HMM+SAAS	11.75%
SAAS+SRC+MV	14.98%
GRAD (SAAS+SRC+GR)	10.11%

From Table VI, we have the following observations.

- 1) The TFER rate for the HMM method is the worst. The main reason is that the HMM method does not differentiate symmetric subgroups inside the asymmetric group. Instead it directly calculates group features over the whole asymmetric group. This makes it unable to capture the asymmetric interactions between members inside the group. Compared with the HMM method, the other three methods, which perform asymmetric activity recognition based on our proposed SAAS, have better performance. This demonstrates the effectiveness of our SAAS. It should be noted that it is possible to develop better features than the ones in Table III to improve the performance of the HMM method for this experiment. However, our SAAS is still important because (a) when the number of group members becomes larger, the interactions between members may be very complicated. It will be very difficult to develop good features for the whole group without considering its lower level structures. (b) In many applications, people are interested in not only the behavior of the whole group but also the behavior of each individual person or subgroups of people. In this case, the HMM method will require a large number of separate models for each individual person or subgroups while our SAAS can do all the tasks in one framework.
- 2) The performance of the GRAD method is better than the SAAS+SRC+MV method. This will be further demonstrated in later experiments.
- 3) The performance of the GRAD method, which uses P-GR, is slightly better than the Layered HMM+SAAS method. Since we calculate the extra features of the higher layer HMM by taking the average of people in each symmetric subgroup, the Layered HMM+SAAS method can be viewed as an extension of using the V-GR. Therefore, the result further implies that P-GR can improve the results from V-GR by discarding the outliers from recognition. Since both the GRAD method and the Layered HMM+SAAS method can recognize hierarchical structure activities, we will discuss more of these two methods in the following section.

C. Experimental Results for Recognizing Hierarchical-Structure Activities with a Fixed Number of Group Members

In this section, we perform experiments to recognize *hierarchical structure activities* which contain four symmetric activities (*InGroup*, *WalkTogether*, *Fight*, and *RunTogether*) and three asymmetric activities (*Approach*, *Split*, and *Chase*).

TABLE VII

TFER COMPARISON FOR HIERARCHICAL-STRUCTURE ACTIVITY RECOGNITION WITH FIXED NUMBER OF GROUP MEMBERS

Methods		TFER	
		Symmetric Activity	Asymmetric Activity
Layered HMM+SAAS		5.36%	11.75%
SAAS+SRC+MV	SAAS+SRC+MV-1	5.52%	14.98%
	SAAS+SRC+MV-2	5.36%	
GRAD (SAAS+SRC+GR)	GRAD-1	5.52%	10.11%
	GRAD-2	5.36%	

Similar to the previous experiment, we fix the number of people in each asymmetric group as 4, and each asymmetric group contains two symmetric subgroups with one group containing three people and the other group containing one person. For simplification, we only recognize the symmetric activity of the three-people subgroup and the asymmetric activity of the four people group in this experiment.

As mentioned in Table IV, the HMM method cannot recognize hierarchical-structure activities. Therefore, we only compare the other three methods. Table VII shows the results for hierarchical-structure activity recognition under 50% training and 50% testing.

Since the numbers of group members are the same as the previous experiments, the TFER results for symmetric activities and asymmetric activities in Table VII are exactly the same as those in Tables V and VI, respectively. We can see from Table VII that the GRAD method and the Layered HMM+SAAS method have similar performance. However, compared with the Layered HMM+SAAS method as well as other HMM-based methods [15]–[19], our proposed GRAD method has the following advantages.

- 1) The Layered HMM+SAAS method as well as most other HMM-based methods [2], [13], [16], [19] cannot handle the recognition with a varying number of group members while our GRAD algorithm can handle this problem by the use of the GR.
- 2) More importantly, there may be hierarchical-structure activities with more than two levels. For example, several asymmetric groups may form a super symmetric group and these super symmetric groups may further form an even larger asymmetric group. In these cases, the HMM-based methods may require very complicated models for recognition which may be very difficult for training and calculation. However, since our GRAD method only extracts GRs from the groups for the recognition in the higher level, it can be kept simple even for those multilevel-structure activities.

D. Experimental Results for Recognizing Hierarchical-Structure Activities with a Varying Number of Group Members

In the above sections, we have demonstrated that our GRAD algorithm has comparable or better results than the previous methods when handling the special scenarios that the previous algorithms can also handle. In this section, we will perform

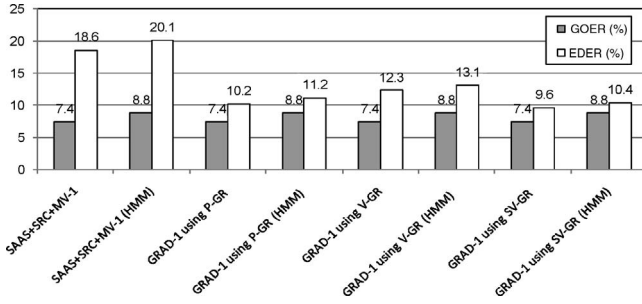


Fig. 7. Experimental results for hierarchical-structure activity recognition with varying number of group members.

experiments for the general scenario of *hierarchical-structure activities* with a *varying number of group members* and try to recognize all of the group activities in Table I for all symmetric and asymmetric groups. From Table IV, we can see that only the SAAS+SRC+MV method and the GRAD method can handle the task in this experiment. Therefore, we only compare these two methods in this section.

In this experiment, we randomly select three long sequences for training and use the other three long sequences for testing. Five independent experiments are performed and the results are averaged.

The experimental results of SAAS+SRC+MV-1 and GRAD-1 are shown in Fig. 7. For the GRAD method, three different GRs are used: 1) physical GR (P-GR in Fig. 7); 2) virtual GR (V-GR in Fig. 7); and 3) selective Virtual GR (SV-GR in Fig. 7). In order to show the advantage of using AHMM, we also include the results of using regular HMM [22] for modeling the activity correlation metric [with ‘‘HMM’’ in Fig. 7, e.g., SAAS+SRC+MV-1 (HMM)].

In order to take clustering errors into consideration, two error rates are compared in Fig. 7: the group clustering error rate (GCER) and the event detection error rate (EDER). They are defined in (12) and (13), respectively

$$GCER = \# \text{ of clustering error frames} / \# \text{ of total frames} \quad (12)$$

$$EDER = \# \text{ of error frames} / \# \text{ of total frames} \quad (13)$$

where a frame is a clustering error frame if any person in the frame is mis-clustered into another symmetric group, and a frame is an error frame if any of the following take place: 1) any person in the frame is mis-clustered into another symmetric group; 2) any of the symmetric activities is misclassified; and 3) any of the asymmetric activities is misclassified.

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

Several observations from Fig. 7 are listed below.

Since all methods use the proposed SRC clustering algorithm for clustering people into symmetric groups, their GCERs are the same if using the same activity-correlation-metric model. Therefore, we can see from Fig. 7 that the GCERs of SAAS+SRC+MV-1, GRAD-1 using P-GR,

GRAD-1 using V-GR, and GRAD-1 using SV-GR are the same. Similarly, the GCERs of SAAS+SRC+MV-1 (HMM), GRAD-1 using P-GR (HMM), GRAD-1 using V-GR (HMM), and GRAD-1 using SV-GR (HMM) are the same. The low GCER demonstrates the effectiveness of the SRC clustering algorithm. Furthermore, methods using AHMM as an activity-correlation-metric model have a better GCER than those use HMM. This demonstrates that using AHMM can improve the performance by handling the possible action asynchronies.

Comparing the EDER, we can see that the EDERs of the GRAD algorithm are obviously better than those of using MV. This supports our claim that the introduction of GR can greatly improve the detection rate for asymmetric activities. Comparing the three GR-based methods, we can see that the EDER of P-GR is better than that of V-GR. This further demonstrates that the P-GR can improve the performance by discarding outliers from asymmetric activity recognition. However, the EDER difference between these two GRs is not large. This is because (a) although V-GR includes outliers, the effect of these outliers is decreased by the averaging with non-outliers, and (b) there may be cases where none of the actual person in the symmetric group is representative enough for the group, and in these cases, the P-GR may not perform better than the V-GR. Furthermore, the method using SV-GR has the best EDER. This is because SV-GR has the following two advantages: 1) similar to P-GR, SV-GR can discard outliers by averaging only the few most representative people in the group, and 2) in case when there is no actual person representative for the group, SV-GR can create a virtual GR by averaging several people in the group. However, we can also see from Table VIII that the improvement of SV-GR from P-GR is small. This is for two reasons. 1) The clustering errors (i.e., GCER) take a large portion of the errors in EDER. This limits the improvement space of SV-GR. It is expected that the performance of the GRAD algorithm can be further improved if people can be clustered better into symmetric groups. 2) Due to the scenarios of the BEHAVE dataset, people in each symmetric subgroup are comparatively close to each other, therefore the chances that none of the actual person is representative are low.

Fig. 8 shows the average false alarm rate (FA) and miss detection rate (Miss) [9] of the GRAD algorithm for the activities in Table I, where the SV-GR is used for the GRAD algorithm. The Miss rate is defined by $\frac{cnt_{\theta}^{fn}}{cnt_{\theta}^{+}}$, where cnt_{θ}^{fn} is the number of false negative (mis-detection) samples for activity θ , and cnt_{θ}^{+} is the total number of positive samples of activity θ in the test data. The FA rate is defined by $\frac{cnt_{\theta}^{fp}}{cnt_{\theta}^{-}}$, where cnt_{θ}^{fp} is the number of false positive (false alarm) samples for activity θ , and cnt_{θ}^{-} is the total number of negative samples of activity k in the test data.

From Fig. 8, we can see our GRAD algorithm have good performance in recognizing most activities. However, the Miss rate for some activities such as *Fighting* and *Chase* are still high. This is because 1) the input features are very simple which are all derived from the MBB information; 2) the number of training samples for these activities is small; and 3) it is more difficult to correctly cluster the symmetric activities

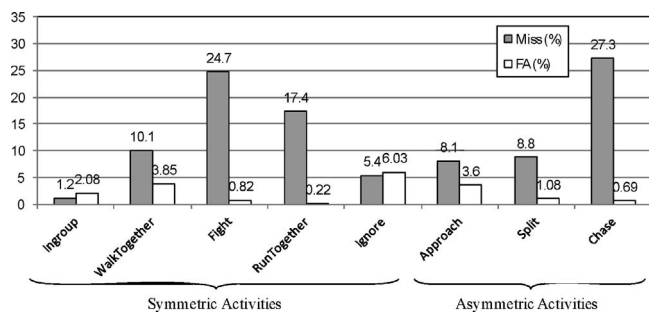


Fig. 8. Average frame level FA and Miss for the GRAD algorithm.

such as *Fighting* due to their large variance. Therefore, in order to further improve the performance, more sophisticated input features [19], [20] can be used and the methods to train models in case of insufficient training data can be introduced [9], [25]. Furthermore, Fig. 8 also shows a large FA rate in the activity *Ignore*. This is because *Ignore* is a generalized activity in our experiment. Since we model *Ignore* as the non-interaction case between people, it can be confused with all the other activities including both symmetric and asymmetric ones. This leads to the large number of samples misclassified as *Ignore*.

VII. ALGORITHM EXTENSION

In the GRAD algorithm proposed in this paper, we model hierarchical-structure activities based on our SAAS and cluster people into symmetric subgroups based on the SRC clustering algorithm. The higher level asymmetric activities between symmetric subgroups can then be recognized based on the interactions between GRs for each symmetric subgroup. We believe that the framework of our proposed GRAD algorithm is general and can easily be extended. In this section, we discuss some possible extensions of our GRAD algorithm.

- 1) In this paper, we use SAAS to model hierarchical activities as a two-level structure with symmetric activities as the lower level and asymmetric activities as the higher level. This two-level structure can cover many scenarios in daily life. However, as mentioned, there may be activities with other hierarchical structures. For example, one *approaching* group may *chase* another *splitting* group and these two asymmetric groups will form a *super asymmetric group*. In these cases, we can extend our GR method so that GRs can also be calculated and used to represent asymmetric groups. Furthermore, we can also extend our SAAS to model different activity structures. In the above example, we can first extend our SAAS by adding one more asymmetric activity level over the original asymmetric level to form a symmetric–asymmetric–asymmetric activity structure. The *chase* activity can then be recognized based on the interactions between the two GRs of the two asymmetric subgroups of *approaching* and *splitting*.
- 2) In the experiments of this paper, all asymmetric activities take place only between two symmetric subgroups. However, there may be cases that the asymmetric activities take place among three or more entities. For exam-

ple, person *A* is *approaching* the symmetric subgroup *B*, at the same time, another person *C* is also *approaching* group *B* from another direction. These three symmetric subgroups *A*, *B*, and *C* will form an asymmetric group of *approaching*. In these cases, we can extend our SRC clustering method to further cluster symmetric subgroups into asymmetric groups. In the above example, we can first calculate the *distance metrics* between *A*, *B*, and *C* based on their asymmetric interaction, and then cluster them into one asymmetric group.

- 3) In this paper, we use AHMM to model the activity correlation metric between any two people, use our SRC clustering method to cluster people into symmetric subgroups, and use one of the three proposed GRs (P-GR, V-GR, and SV-GR) to represent each symmetric subgroup. However, since the framework of our GRAD algorithm is general, other models, clustering methods, and GR calculation methods can also be used to improve the performance of the GRAD method.

VIII. CONCLUSION

In this paper, we proposed 1) an SAAS for the detection of hierarchical activities; 2) a GR to handle group event detection with a varying number of group members; and 3) an SRC clustering algorithm to deal with clustering with an asymmetric distance metric. Experimental results demonstrate the effectiveness of our proposed algorithm.

ACKNOWLEDGMENT

The authors would like to thank Dr. S. 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, no. 3, pp. 509–520, Jun. 2006.
- [2] N. Oliver, A. Garg, and E. Horvitz, "Layered representations for learning and inferring office activity from multiple sensory channels," *Comput. Vis. Image Understand.*, vol. 96, no. 2, pp. 163–180, 2004.
- [3] S. Park and J. K. Aggarwal, "A hierarchical Bayesian network for event recognition of human actions and interactions," *Multimedia Syst.*, vol. 10, no. 2, pp. 164–179, 2004.
- [4] S. Hongeng and R. Nevatia, "Multiagent event recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2, 2001, pp. 84–91.
- [5] N. Vaswani, A. R. Chowdhury, and R. Chellappa, "Activity recognition using the dynamic of the configurations of interacting objects," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 2, 2003, pp. 633–640.
- [6] D. Wyatt, T. Choudhury, and J. Bilmes, "Conversation detection and speaker segmentation in privacy-sensitive situated speech data," in *Proc. Interspeech*, 2007, pp. 586–589.
- [7] S. Bengio, "An asynchronous hidden Markov model for audio-visual speech recognition," in *Proc. 15th Adv. NIPS*, 2003, pp. 1237–1244.
- [8] *BEHAVE Dataset* [Online]. Available: <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. Circuits Syst. Video Technol.*, vol. 18, no. 8, pp. 1128–1139, Aug. 2008.
- [10] H. Späth, *Cluster Analysis Algorithms for Data Reduction and Classification of Objects*. New York: Halsted, 1980.
- [11] K. Smith, D. Gatica-Perez, and J. M. Odobez, "Using particles to track varying number of interacting people," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 1, 2005, pp. 962–969.

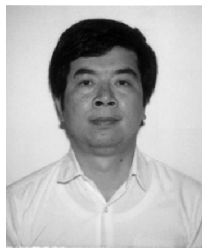
- [12] L. R. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
- [13] T. V. Duong, H. H. Bui, D. Q. Phung, and S. Venkatesh, "Activity recognition and abnormality detection with the switching hidden semi-Markov model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, vol. 1. 2005, pp. 838–845.
- [14] B. Li, E. Chang, and C. T. Wu, "DPF—a perceptual distance function for image retrieval," in *Proc. Int. Conf. Image Process.*, vol. 2. 2002, pp. 597–600.
- [15] F. Lv, J. Kang, R. Nevatia, I. Cohen, and G. Medioni, "Automatic tracking and labeling of human activities in a video sequence," in *Proc. IEEE Workshop Performance Evaluat. Track. Surveillance*, 2004, pp. 33–40.
- [16] I. McCowan, D. Gatica-Perez, S. Bengio, G. Lathoud, M. Barnard, and D. Zhang, "Automatic analysis of multimodal group actions in meetings," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 3, pp. 305–317, Mar. 2005.
- [17] L. Lam and S. Y. Suen, "Application of majority voting to pattern recognition: An analysis of its behavior and performance," *IEEE Trans. Syst., Man Cybernet.*, vol. 27, no. 5, pp. 553–568, Sep. 1997.
- [18] S. B. Oh, "On the relationship between majority vote accuracy and dependency in multiple classifier systems," *Pattern Recognit. Lett.*, vol. 24, pp. 359–363, Jan. 2003.
- [19] Y. Song, L. Goncalves, and P. Perona, "Unsupervised learning of human motion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 7, pp. 814–827, Jul. 2003.
- [20] Y. A. Ivanov and A. F. Bobick, "Recognition of visual activities and interactions by stochastic parsing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 852–872, Aug. 2000.
- [21] A. Amer, "Voting-based simultaneous tracking of multiple video objects," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 11, pp. 1448–1462, Nov. 2005.
- [22] J. Bilmes, "A gentle tutorial of the EM algorithm and its application to parameter estimation for gaussian mixture and hidden Markov models," Dept. Electric. Eng. Comput. Sci., Univ. California, Berkeley, Tech. Rep. ICSI-TR-97-021, 1997.
- [23] P. C. Ribeiro and J. Santos-Victor, "Human activity recognition from video: Modeling, feature selection and classification architecture," in *Proc. Int. Workshop Human Activity Recognit. Model.*, 2005, pp. 61–70.
- [24] B. Moghaddam and A. Pentland, "Probabilistic visual learning for object representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 696–710, Jul. 1997.
- [25] W. Lin, M.-T. Sun, R. Poovendran, and Z. Zhang, "Human activity recognition for video surveillance," in *Proc. Int. Symp. Circuits Syst.*, 2008, pp. 2737–2740.



Weiyao Lin received the B.Eng. and M.Eng. degrees from Shanghai Jiao Tong University, Shanghai, China, in 2003 and 2005, respectively, and the Ph.D. degree from the University of Washington, Seattle, in 2010, all in electrical engineering.

Since 2010, he has been an Assistant Professor with the Department of Electronic Engineering, Institute of Image Communication and Information Processing, Shanghai Jiao Tong University. His current research interests include video processing, machine learning, computer vision, and video coding

and compression.



Ming-Ting Sun (S'79–M'81–SM'89–F'96) received the B.S. degree from National Taiwan University, Taipei, Taiwan, in 1976, the M.S. degree from the University of Texas, Arlington, in 1981, and the Ph.D. degree from the University of California, Los Angeles, in 1985, all in electrical engineering.

Since 1996, He has been with the University of Washington, Seattle, where he is currently a Professor with the Department of Electrical Engineering. Previously, he was the Director of the Video Signal Processing Research Group, Bellcore, Red Bank, NJ.

He was a Chaired Professor with TsingHwa University, Beijing, China, and Visiting Professor with Tokyo University, Tokyo, Japan, and National Taiwan University. He holds 11 patents and has published over 200 technical papers, including 13 book chapters in the area of video and multimedia technologies.

He co-edited the book, *Compressed Video Over Networks* (New York: Marcel Dekker, 2001).

Dr. Sun was the Editor-in-Chief of the IEEE TRANSACTIONS ON MULTIMEDIA (TMM) and a Distinguished Lecturer of the Circuits and Systems Society from 2000 to 2001. He received the IEEE CASS Golden Jubilee Medal in 2000, and was the General Co-Chair of the Visual Communications and Image Processing 2000 Conference. He was the Editor-in-Chief of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (TCSVT) from 1995 to 1997. He received the TCSVT Best Paper Award in 1993. From 1988 to 1991, he was the Chairman of the IEEE CAS Standards Committee and established the IEEE Inverse Discrete Cosine Transform Standard. He received the Award of Excellence from Bellcore for his work on the digital subscriber line in 1987.



Radha Poovendran (SM'06) received the Ph.D. degree in electrical engineering from the University of Maryland, College Park, in 1999.

He is an Associate Professor and Founding Director of the Network Security Laboratory with the Department of Electrical Engineering, University of Washington, Seattle. He is a Co-Editor of the book *Secure Localization and Time Synchronization in Wireless Ad Hoc and Sensor Networks* (Berlin, Germany: Springer Verlag, 2007). His current research interests include applied cryptography for multiuser

environment, wireless networking, and applications of information theory to security.

Dr. Poovendran is the recipient of the NSA LUCITE Rising Star Award, Faculty Early Career Awards including NSF CAREER in 2001, ARO YIP in 2002, ONR YIP in 2004, and PECASE in 2005 for his research contributions to multiuser security, and the Graduate Mentor Recognition Award from the University of California, San Diego, in 2006. He co-chaired the first ACM Conference on Wireless Network Security in 2008. He is the Lead Editor of the upcoming IEEE Proceedings special issue on Cyber-Physical Systems 2011.



Zhengyou Zhang (SM'97–F'05) received the B.S. degree in electronic engineering from the University of Zhejiang, Zhejiang, China, in 1985, the M.S. degree in computer science (specialized in speech recognition and artificial intelligence) from the University of Nancy, Nancy, France, in 1987, the Ph.D. degree in computer science (specialized in computer vision) from the University of Paris XI, Paris, France, in 1990, and the D.Sc. (habilitation a diriger des recherches) Diploma from the University of Paris XI in 1994.

He is a Principal Researcher with Microsoft Research, Microsoft Corporation, Redmond, WA. He has been with INRIA (French National Institute for Research in Computer Science and Control) for 11 years and was a Senior Research Scientist from 1991 until he joined Microsoft Research in 1998. From 1996 to 1997, he spent a one-year sabbatical as an Invited Researcher with Advanced Telecommunications Research Institute International, Kyoto, Japan. He has published over 150 papers in refereed international journals and conferences, and has co-authored the following books: *3-D Dynamic Scene Analysis: A Stereo Based Approach* (Berlin, Germany: Springer, 1992); *Epipolar Geometry in Stereo, Motion and Object Recognition* (Norwell, MA: Kluwer, 1996); *Computer Vision* (textbook in Chinese, Chinese Academy of Sciences, 1998); *Face Geometry and Appearance Modeling: Concepts and Applications* (Cambridge, U.K.: Cambridge Univ. Press, 2010).

Dr. Zhang is the Founding Editor-in-Chief of the IEEE TRANSACTIONS ON AUTONOMOUS MENTAL DEVELOPMENT, an Associate Editor of the *International Journal of Computer Vision*, and an Associate Editor of *Machine Vision and Applications*. He served on the Editorial Board of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE from 2000 to 2004, and the IEEE TRANSACTIONS ON MULTIMEDIA from 2003 to 2008, among others. He has been on the program committees for numerous international conferences, and was an Area Chair and a Demo Chair of the International Conference on Computer Vision, Nice, France, in 2003, a Program Co-Chair of the Asian Conference on Computer Vision, Jeju Island, Korea, in 2004, a Demo Chair of the International Conference on Computer Vision, Beijing, China, in 2005, a Program Co-Chair of the International Workshop of Multimedia Signal Processing, Victoria, Canada, in 2006, and a Program Co-Chair of International Workshop on Motion and Video Computing, Austin, TX, in 2006. He has given a number of keynotes in international conferences.