Activity Recognition Using a Combination of Category Components and Local Models for Video Surveillance

Weiyao Lin, *Student Member, IEEE*, 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 human activities for video surveillance applications. We propose to represent an activity by a combination of category components and demonstrate that this approach offers flexibility to add new activities to the system and an ability to deal with the problem of building models for activities lacking training data. For improving the recognition accuracy, a confident-frame-based recognition algorithm is also proposed, where the video frames with high confidence for recognizing an activity are used as a specialized local model to help classify the remainder of the video frames. Experimental results show the effectiveness of the proposed approach.**

***Index Terms*—Category components, event detection, local model, video surveillance.**

I. Introduction and Related Work

VIDEO surveillance is of increasing importance in many applications, including elder care, home nursing, and unusual event alarming [1]–[4]. Automatic activity recognition plays a key part in video surveillance. In this paper, we focus on addressing the following three key issues for event recognition.

A. Flexibility of the System for Adding New Events

In many applications, people may often want to add new events of interest into the recognition system. It is desirable that the existing models in the system are not affected or do not need to be reconstructed when new events are added. Using most existing activity recognition algorithms [12]–[17], [26], [27], the whole system has to be retrained or reconstructed for the new added events. Some methods [5], [6], [24] try to use a similarity metric so that different events can be clustered into different groups. This approach has more flexibility for new added events. However, due to the uncertain nature of the activity instances, it is difficult to find a suitable feature set for all samples of an event to be clustered closely around a center.

B. Recognition of Events Lacking Training Samples

In many surveillance applications, events of interest may only occur rarely (e.g., most unusual events such as a heart attack or falling down stairs). For these events, it is difficult to collect sufficient training samples for learning the unusual event models. In this case, many event detection algorithms [8], [9] that require large numbers of training data become unsuitable. Methods for learning from small numbers of examples are needed [5]–[7], [26]–[29]. In this paper, we call these lacking-training-sample (LTS) events.

Several algorithms have been proposed to address the difficulty of recognizing LTS events. Wu *et al.* [27] and Amari *et al.* [26] try to solve the unbalanced-training-data problem by using a conformal transform to adapt the support vector machine (SVM) kernels. However, these methods still need boundary training samples (samples around class boundaries) to obtain good support vectors for differentiating different classes, while in reality the insufficient training set may not include these boundary training samples. Other researchers [28], [29] try to improve the estimation of model parameters (e.g., the Gaussian covariance matrix) for cases of limited training samples. However, these methods do not work well if the limited training data are not sufficient to fully represent events.

C. Accuracy for Recognizing Human Activities

Recognition accuracy is always a major concern for automatic event recognition. Many algorithms have been proposed which try to detect human activities with high accuracy. Cristani *et al.* [22], Zhang *et al.* [35], and Dupant *et al.* [36] focus on developing suitable multistream fusion methods to combine features from different streams (e.g., audio and video) to improve the recognition accuracy. Cristani *et al.* [22] propose an AVC matrix for audio and video stream fusion. Dupant *et al.* [36] propose to use Weighted Multiplication for combining multistream data. Zhang *et al.* [35] compare different stream-combining methods such as weighted multiplication and early integration. Models such as hidden Markov model (HMMs) or dynamic Bayesian network (DBN) [12]–[14], state machine [4], [15], Adaboost [16], [17], and SVM [26], [27] are widely used in these works for activity recognition. However, most of these methods only work well in their assumed

Manuscript received November 9, 2007; revised March 7, 2008. First published June 17, 2008; current version published August 29, 2008. This work was supported in part by the Army Research Office under PECASE Grant W911NF-05-1-0491 and MURI Grant W 911 NF 0710287. This paper was recommended by Guest Editor I. Ahmad.

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

Z. Zhang is with Microsoft Research, Microsoft Corporation, Redmond, WA 98052 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.2008.927111

scenarios and have limitations or lower accuracy if applied to other scenarios. Therefore, it is always desirable to develop new algorithms to improve the recognition accuracy.

The contribution of this paper is summarized as follows.

1) To address the flexibility problem for adding new events, we propose to use a category feature vector (CFV)-based model to represent an event.
2) To address the problem of recognizing events which lack training samples (LTS events), we propose a new approach to derive models for the LTS events from the parts from other trained related events.
3) To address the accuracy problem for recognition algorithms, we propose a confident-frame-based recognition algorithm (CFR) to improve the accuracy of the recognition.

The remainder of this paper is organized as follows. Section II describes our approach to represent activities. Based on this activity representation, Section III discusses the flexibility of our method for training new activities. Section IV describes our proposed method to train models for events lacking training data. In Section V, we present our CFR to improve the recognition accuracy. Experimental results are shown in Section VI. We conclude the paper in Section VII.

II. Activity Representation

for flexible classification, activities can be described by a combination of feature attributes. for example, a set of human activities (*inactive*, *active*, *walking*, *running*, *fighting*) [11] can be differentiated using a combination of attributes of two features: *change of body size* (CBS) and *speed*. Each feature can have attributes *high*, *medium*, and *low*. *Inactive*, which represents a static person, can be described as *low CBS* and *low speed*. *Active*, which represents a person making movements but little translations, can be described as *medium CBS* and *low speed*. *Walking*, representing a person making movements and translations, can be described as *medium CBS* and *medium speed*. *running*, which is similar to *walking* but with a larger translation, can be described as *high CBS* and *high speed*. *Fighting*, which has large movements with small translations, can be described as *high CBS* and *low speed*.

It is efficient to represent the activities by the combination of *feature-attributes* as shown in the above example. A relatively small number of features and attributes can describe and differentiate a large number of activities (n features with m attributes could describe m^n activities). However, this approach has low robustness. The misclassification of one feature attribute can easily lead to a completely wrong result. Furthermore, the extent of "*medium*," "*low*," or "*high*" is difficult to define.

The above *feature-attribute* description for representing activities can be extended to a description by a combination of CFVs with each CFV F_i containing a set of correlated features. F_i is defined by $F_i = [f_{1i}, f_{2i}, \ldots f_{mi}]^{\mathrm{T}}$, where $f_{1i}, f_{2i}, \ldots f_{mi}$ are correlated features related to the *same* category i. Each CFV can be further represented by different models. For example, using a Gaussian mixture model (GMM) [11], [31], [32] as

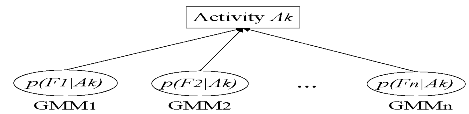


Fig. 1. Activity $\mathrm{A_k}$ is described by a combination of GMMs with each GMM representing the distribution $p(F_i|A_k)$ of a CFV F_i.

shown in Fig. 1, the likelihood function $p(F_i(t)|A_k)$ of the observed CFV $F_i(t)$ for video frame t, given activity A_k, can be described as

$$p\left(F_i(t)|A_k\right) \approx \sum_{j=1}^{M_{\mathrm{F}i,\mathrm{A}k}} \pi_j^{\mathrm{F}i,\mathrm{A}k} \cdot N\left(\mu_j^{\mathrm{F}i,\mathrm{A}k}, \sigma_j^{\mathrm{F}i,\mathrm{A}k}\right) \qquad (1)$$

where $\pi_j^{Fi,Ak}$ is the weight of the jth Gaussian distribution $N(\mu_j^{Fi,Ak}, \sigma_j^{Fi,Ak})$ for the CFV of $F_i(t)$ given activity A_k. $\mu_j^{Fi,Ak}$ and $\sigma_j^{Fi,Ak}$ are the mean and variance for distribution $N(\mu_j^{Fi,Ak}, \sigma_j^{Fi,Ak})$, respectively. $\pi_j^{Fi,Ak}$ is normalized to make $p(F_i(t)|A_k)$ a proper probability distribution. $M_{Fi,Ak}$ is the number of Gaussian mixtures for F_i given A_k.

Essentially, CFV is the extension of the "feature" in the *feature-attribute* description. Features with high correlations for describing activities are grouped into the same CFV. The GMM model $p(F_i(t)|A_k)$ is the extension of the "feature attribute" in the *Feature-Attribute* description. With the use of the CFV representation, we will have more robustness in representing and recognizing activities compared with the *Feature-Attribute* description. It should be noted that although in this paper we use a GMM to represent a CFV, the proposed CFV representation is not limited to the GMM model. Other models such as HMM or DBN can also be used to represent a CFV.

In practice, CFVs can be formed by clustering features based on feature similarities such as *correlation coefficient* [30] or *K-L distance* [10], [25]. In the experiments presented in this paper, the CFVs are formed by clustering the features based on their K-L distances. The similarity of two feature distributions can be approximated by the K-L distance in terms of the means and variances of the Gaussian distributions [10], [25]

$$D(f_i, f_j) = \sum_{k \in A} \left\{ (\mu_{i,k} - \mu_{j,k})^2 \cdot \left(\frac{1}{\sigma_{i,k}^2} + \frac{1}{\sigma_{j,k}^2} \right) + \frac{\sigma_{i,k}^2}{\sigma_{j,k}^2} + \frac{\sigma_{j,k}^2}{\sigma_{i,k}^2} \right\} \qquad (2)$$

where f_i, f_j are two features, and $\mu_{i,k}$ and $\sigma_{i,k}$ are the mean and the variance of the probability distribution $p(f_i|A_k)$. By grouping correlated features into a CFV, the correlations of the features can be captured by the GMM. Also, we can reduce the total number of GMM models, which can facilitate the succeeding classifier which is based on fusing the GMM results. Furthermore, by separating the complete set of features into CFVs, it facilitates the handling of new added activities and the training of models for LTS events as described below.

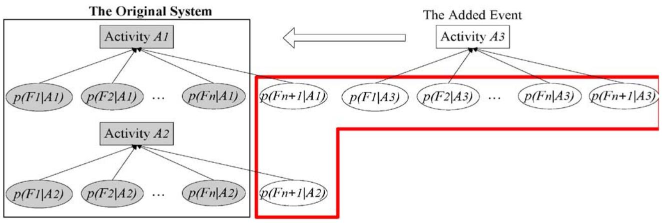


Fig. 2. Flexibility for adding a new event A_3. (Gray circles: models do not need to be changed; white circles: models need training).

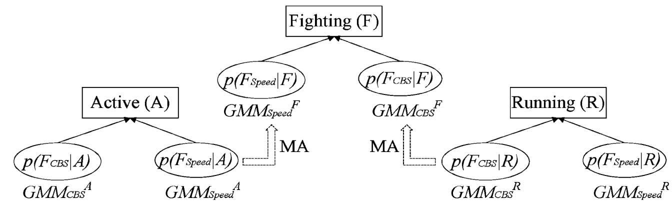


Fig. 3. Training of an LTS event *Fighting*.

III. Handling New Added Activities

When new activities are added to the system, the already defined features may not be enough to differentiate all activities, necessitating the adding of new features. With our CFV-based representation, we only need to define new categories for the added features (i.e., define new CFVs) and train new models for them (i.e., add a new GMM for each new CFV), while keeping the other CFV-GMMs of the already existing events unchanged. For example, in Fig. 2, the original system has two activities A_1 and A_2, each activity has n CFV-based GMM models to represent it (gray circles in Fig. 2). When a new activity A_3 is added to the system, new features are needed to differentiate the three activities. We define a new CFV F_{n+1} with a GMM model $p(F_{n+1}|A_k)$ for these new features and add it to all activities A_k. The flexibility of our representation is that we only need to train new models for the new event A_3 as well as the newly added model $p(F_{n+1}|A_k)$ for the existing events A_1 and A_2 (white circles in the bold rectangle in Fig. 2), while keeping all the existing models in the original system unchanged (grey circles in Fig. 2). In practice, the number of trained activities could be much larger compared with the number of new added events. The flexibility offered by the CFV-based system enables the possibility of designing a programmable system which can incrementally grow, instead of needing to retrain the whole system when a new event needs to be added. In contrast to the above, the models of traditional methods will become increasingly complicated with the addition of new features.

IV. Training Models for LTS Events

Since LTS events lack training data, we often do not have enough data to construct a representative model for these events. To solve this problem, we observe that people often describe a rare object by a combination of different parts from familiar objects. For example, people may describe a llama as an animal with a head similar to a camel and a body similar to a donkey. Similarly, with our CFV-based representation of activities, it is possible for us to derive a good initial LTS event model from the CFVs of the previously trained activities. For example, as shown in Fig. 3, we have trained two CFVs F_{CBS} and F_{Speed} for recognizing four events: *Active*, *Inactive*, *Walking*, and *Running*. F_{CBS} is the CFV for the category *CBS*, and F_{Speed} is the CFV for the category *Speed*. Assume *Fighting* is an event we try to recognize but lacking training data. For the CBS category, we can reason that the behavior of *Running* is the most similar among all of the usual events to that of *Fighting*, therefore, the GMM for $F_{\text{CBS}}^{\text{fighting}}$ will be adapted from that of $F_{\text{CBS}}^{\text{running}}$. Similarly, for the Speed Category, we find that the behavior of *Active* is the most similar to that of *Fighting*, therefore, the GMM for $F_{\text{Speed}}^{\text{fighting}}$ will be adapted from that of $F_{\text{Speed}}^{\text{active}}$. In this way, we can have a good initial model for *Fighting* even if we lack training data.

We propose to generate models for LTS activities as follows. For each CFV_i in category i of the LTS activity A_u, find the trained GMM model (GMM_i^{Ak}) where the behavior of activity A_k is most similar to the LTS activity A_u in this *specific* category.

This initial model can be adapted further to derive a new model GMM_i^{Au} using the limited training data and the MAP-based adaptation (MA) [7], [33]. MA is an extension of the EM algorithm which contains two steps.

Step 1) Update the parameters θ^{New} by the regular EM algorithm [34] with the limited training data.

Step 2) The GMM parameters are then adapted by the linear combination of the parameters θ_{New} and the parameters of the initial model θ^{Old} (the parameters of GMM_i^{Ak}):

$$\begin{cases} w_i^A = \alpha \cdot w_i^{\text{old}} + (1-\alpha) \cdot w_i^{\text{new}} \\ \mu_i^A = \alpha \cdot \mu_i^{\text{old}} + (1-\alpha) \cdot \mu_i^{\text{new}} \\ \sigma_i^A = \alpha \cdot \left(\sigma_i^{\text{old}} + \left(\mu_i^A - \mu_i^{\text{old}}\right)\left(\mu_i^A - \mu_i^{\text{old}}\right)^T\right) \\ \quad +(1-\alpha) \cdot \left(\sigma_i^{\text{new}} + \left(\mu_i^A - \mu_i^{\text{new}}\right)\left(\mu_i^A - \mu_i^{\text{new}}\right)^T\right) \end{cases} \tag{3}$$

where $\{w_i^A, \mu_i^A, \sigma_i^A\}$ are the weight, mean, and variance of the adapted model for the ith Gaussian in the GMM, $\{w_i^{\text{old}}, \mu_i^{\text{old}}, \sigma_i^{\text{old}}\}$ are the parameters of the initial model θ^{Old}, and $\{w_i^{\text{new}}, \mu_i^{\text{new}}, \sigma_i^{\text{new}}\}$ are the updated parameters from the regular EM algorithm in Step 1). α is the weighting factor to control the balance between the initial model and the new estimates.

V. Confident-Frame-Based Recognition Algorithm

After the activities are described by the combination of CFV-based GMMs, we can construct a GMM classifier C_i for each CFV F_i with the MAP principle, as shown as

$$\begin{cases} C_i\left(F_i(t)\right) = \arg\max_{Ak} P\left(A_k|F_i(t)\right) \\ P\left(A_k|F_i(t)\right) = \frac{p(F_i(t)|A_k)\cdot P(A_k)}{p(F_i(t))} \end{cases} \tag{4}$$

where $p(F_{\text{i}}(t)|A_k)$ is the likelihood function for the observed CFV of $F_i(t)$ in category i at frame t, given activity A_k, calculated by (2). $P(A_k)$ is the probability for activity A_k and $p(F_{\text{i}}(t))$ is the likelihood function for the CFV $F_i(t)$.

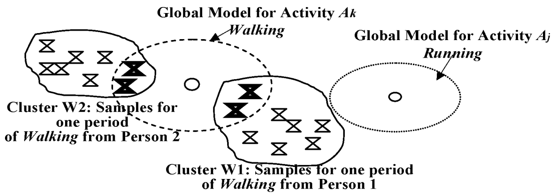


Fig. 4. Global and local models.

The GMM classifiers for different CFVs will differentiate activities with different confidence (e.g., the classifier C_{CBS} is more capable to differentiate *Inactive* and *Fighting*, while the classifier C_{Speed} may have more difficulty in doing so), leading to various possible inconsistencies among results from classifiers for different CFVs. Thus, it is desirable to fuse the classification results from different classifiers to obtain the final improved result. In the following, we propose a CFR to improve the recognition accuracy.

A. Combining the Global Model and Local Model for Improved Recognition

Due to the uncertain nature of human actions, samples of the same action may be dispersed or clustered into several groups. The "global" model derived from the whole set of training data collected from a large population of individuals with significant variations may not give good results in classifying activities associated with an individual. In this section, we introduce the idea of using local models to improve the accuracy of recognizing activities.

Using Fig. 4 as an example, there are two global models: A_k for activity *walking* and A_j for activity *running*. The cross samples in the figure are frame positions in the feature space with each cluster of crosses representing one period of action taken by one person. Due to the nonrigidness of human actions, each individual person's activity pattern may be "far" from the "normal" patterns of the global model. In this example, if Person 1 walks (cluster $W1$ in Fig. 4) faster than normal people and Person 2 walks (cluster $W2$ in Fig. 4) slower than normal people, then most of the samples in both clusters will be "far" from the center of the "global" model for A_k. When using the global model to perform classification on Cluster $W1$, only a few samples in $W1$ can be correctly classified. The other samples in $W1$ may be misclassified as A_j. However, based on the self-correlation of samples within the same period of action, if we use those samples that are well recognized by the global model (boldfaced crosses in Fig. 4) to generate "local" models, it could greatly help the global model to recognize other samples.

Based on the idea described in the above example, our proposed CFR algorithm can be described as follows.

1) *For an activity A_k, we use the "global" model to detect frames T_k which have high confidence for recognizing A_k, instead of trying to match every frame directly using the global model*. We call T_k *Confident Frames*, while the rest of the frames are called *Left Frames* (denoted as L_k) as shown in Fig. 5. Many methods can be used to detect *confident frames*, such as weighted average [37], [38] or weighted multiplication [36] of recognition results from the CFVs. In this paper, weighted average is used to detect *confident frames* as

$$t = \begin{cases} T_k, & \text{if } \sum_i w_{k,i} \cdot P\left(A_k | F_i(t)\right) > th_k & (5) \\ L_k, & \text{if } \sum_i w_{k,i} \cdot P\left(A_k | F_i(t)\right) \leq th_k & (6) \end{cases}$$

where t is the current frame, T_k is a *Confident Frame*, and L_k is a *Left Frame*. $P(A_k|F_i(t))$ is the recognition result from the global model of CFV F_i. $P(A_k|F_i(t))$ can be calculated by (4). $w_{k,i}$ is the weight for the global model result of CFV F_i under action A_k. $th_k(> 0)$ is the threshold for detecting *confident frames* for A_k. $w_{k,i}$ $(0 \leq w_{k,i} \leq 1, \Sigma_{\text{i}} w_{k,i} = 1)$ and th_k can be selected by the fivefold cross-validation method [35].

2) *The confident frames will be used to generate a "local" model (or specialized model) for the current period of activity A_k. The local model and global model will be used together to classify the Left Frames L_k*. The *global model* is used first to select the most possible candidate events, and then the *local model* is used to decide the best one among the candidate events. The decision on the best candidate event is based on our proposed multi-category dynamic partial function (M-DPF) which is extended from the dynamic partial function (DPF) [21]. The M-DPF is described by

$$D_w(X,Y) = k_1 \cdot \left(\sum_{f_{i1} \in F_1 \setminus \Delta} w_{i1} \cdot \left(f_{i1}^X - f_{i1}^Y\right)^r \right)^{\frac{1}{r}} + \ldots + k_n \cdot \left(\sum_{f_{\text{in}} \in F_n \setminus \Delta} w_{\text{in}} \cdot \left(f_{\text{in}}^X - f_{\text{in}}^Y\right)^r \right)^{\frac{1}{r}} \quad (7)$$

where X and Y are two feature sets. $F_i \setminus \Delta$ are features in F_i but not in Δ. w_{ij} is the weight for the ith feature f_{ij} in CFV F_j. k_j is the weight for CFV of F_j. r is a constant parameter,

$$\Delta = \left\{ \text{the largest } n\ (f^X - f^Y)'s \text{ of the set of } \left[F_1^X - F_1^Y, F_2^X - F_2^Y, \ldots, F_n^X - F_n^Y\right] \right\}$$

and $F_j = [f_{1j}, f_{2j}, f_{3j}, \ldots f_{mj}]^{\text{T}}$ is the CFV for category j. The M-DPF in (7) is used to measure the dissimilarity between the *confident frame* T_k with the feature set X and *left frame* L_k with the feature set Y (the testing sample). Since frames during an activity A_k represent the *same consistent* action of the *same* person, the self-correlation between the frames during A_k should be higher than the correlation between the frames inside the duration of A_k and the frames outside the duration of A_k. This means that normally L_k will be more similar to T_k than T_j if $j \neq k$, as shown in Fig. 5.

B. Summary of the CFR Process

The CFR process is summarized as follows.

Step 1) For a given video sequence, first detect all *confident frames* associated with each activity by (5).

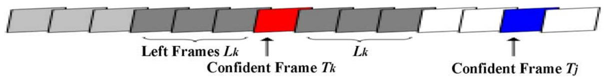


Fig. 5. Confident Frames and Left Frames associated with an activity A_k.

Step 2) For each *Left Frame* t_L, pick the two most possible candidate activities for this frame by

$$\begin{cases} candi\ 1 = \arg\max \sum_i w_{k,i} \cdot P\left(A_k|F_i(t)\right) \\ candi\ 2 = \underset{k,k\neq candi\ 1}{\arg\max} \sum_i w_{k,i} \cdot P\left(A_k|F_i(t)\right) . \end{cases} \quad (8)$$

Step 3) Select the two confident frames T_{candi1} and T_{candi2} corresponding to the two most possible candidate activities which are temporally closest to t_L. If we cannot find T_{candi} in the duration of the activity associated with the current object, the temporally closest T_{candi} of a different object with the same activity can be selected. Then calculate the dissimilarity of $D_w(T_{\text{candi1}}, t_L)$ and $D_w(T_{\text{candi2}}, t_L)$ by (7).

Step 4) The candidate with smaller D_w will be the result for frame t_L.

In the above process, the global model and the local model are used together for classifying the *left frames* in order to increase the accuracy of the recognition. The global model based on the GMMs is first used to select the two most possible candidate activities, then the local model (confident frame-based dissimilarity checking) is used to classify a left frame into one of the two candidate activities.

C. Discussion of CFR and Comparison With Other Methods

Since the CFR method can be considered as a method for combining the results from the CFV classifiers, it can be compared to other multisteam fusion (MF) methods. Compared with most MF methods [22], [35]–[38] or other event detection methods [10]–[17] described in Section I, the major difference of our proposed CFR algorithm is the introduction of the local models to help recognize the *left frames*. With the introduction of local models, the CFR algorithm has the following four advantages.

1) Most MF methods and other methods focus on detecting the occurrence of events and are normally not good at detecting the boundary between two actions precisely, while our CFR method can effectively detect the starting and ending point of activities.
2) In cases when it fails to detect any confident frame during period P_{Ak} of action A_k, CFR may still be able to detect the event by checking the dissimilarity with local models (confident frames) of A_k outside P_{Ak}, as in Fig. 6. This makes it more robust and accurate compare to MF methods.
3) Many MF methods [35]–[38] need to carefully select the fusion parameters in order for these methods to perform well on each sample in the test set. This parameter selection will become more difficult when the number of samples or activities increases. However, CFR only requires the parameters to work well with the local model (confident frames) for each activity period, which will greatly facilitate the parameter selection process.
4) By introducing the local model into the activity recognition, we can also take the advantage of using more features. Some kinds of features such as *object location* may not be suitable for differentiating activities for the classifiers. For example, many activities can take place anywhere, therefore *object location* is not able to differentiate them. However, when checking dis-similarities between the *Confident Frames* and the *Left Frames*, these features will be useful. Therefore, CFR enables the inclusion of more features to facilitate the recognition.

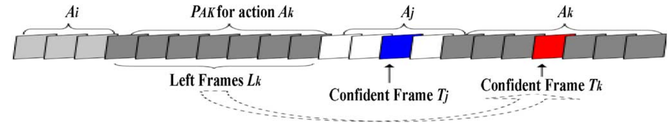


Fig. 6. When failing to detect any confident frame during period P_{Ak}, CFR may still be able to detect the event by checking the dissimilarity with confident frames of A_{k} outside P_{Ak}.

TABLE I
NUMBER OF POSITIVE AND NEGATIVE SAMPLES (VIDEO FRAMES) FOR EACH ACTIVITY

	# of Positive Samples	# of Negative Samples
Inactive	9077	18276
Active	3236	24117
Walking	14144	13209
Running	490	26863
Fighting	406	26947

VI. EXPERIMENTAL RESULTS

In this section, we show experimental results for our proposed methods. The experimental results of the *CFR algorithm to improve recognition accuracy* are shown in Section VI-A. In Section VI-B, experimental results are shown to justify the effectiveness of the proposed *LTS event training method*. Section VI-C shows the results to justify *the flexibility of our algorithm to add new events*.

A. Experimental Results for the CFR Algorithm

We perform experiments using the PETS'04 database [20], and try to recognize five activities: *Inactive*, *Active*, *Walking*, *Running*, and *Fighting*. The total numbers of video frames for each activity are listed in Table I.

For simplicity, we only use the minimum bounding box (MBB) information (which is the smallest rectangular box that includes the object [11]) to derive *all* of the features used for activity recognition. Note that the proposed algorithm is not limited to MBB features. Other more sophisticated features [18], [19] can easily be applied to our algorithm to give better results. It should also be noted that, in our experiments, some activities do not have enough training data. The definitions of features used for activity recognition are listed in the third column of Table II. The features are grouped into two CFVs

TABLE II
CFV AND FEATURE DEFINITIONS

CFV	Feature	Definition of f
CFV_{bm}= [$f_{c\_mb\_sz}$, $f_{c\_mb\_wd}$, $f_{c\_mb\_ht}$]	$f_{c\_mb\_sz}$	$\sum_{i=t-k}^{t+k}\left\lvert\frac{w^{i}_{mbb}\cdot h^{i}_{mbb}-w^{i-1}_{mbb}\cdot h^{i-1}_{mbb}}{w^{i-1}_{mbb}\cdot h^{i-1}_{mbb}}\right\rvert \Big/ 2k+1$
	$f_{c\_mb\_wd}$	$\sum_{i=t-k}^{t+k}\left\lvert\frac{w^{i}_{mbb}-w^{i-1}_{mbb}}{w^{i-1}_{mbb}}\right\rvert \Big/ 2k+1$
	$f_{c\_mb\_ht}$	$\sum_{i=t-k}^{t+k}\left\lvert\frac{h^{i}_{mbb}-h^{i-1}_{mbb}}{h^{i-1}_{mbb}}\right\rvert \Big/ 2k+1$
CFV_{bt}= [f_{avg_spd}, f_{avg_vct}, f_{mean_vct}]	f_{avg_spd}	$\sum_{i=t-k}^{t+k}\sqrt{\left(x^{i}_{MBB}-x^{i-1}_{MBB}\right)^2+\left(y^{i}_{MBB}-y^{i-1}_{MBB}\right)^2} \Big/ 2k+1$
	f_{avg_vct}	$\sqrt{\left(\frac{x^{t+k}_{MBB}-x^{t-k}_{MBB}}{2k+1}\right)^2+\left(\frac{y^{t+k}_{MBB}-y^{t-k}_{MBB}}{2k+1}\right)^2}$
	f_{mean_vct}	$\sum_{i=-(k-1)}^{k}\sqrt{\left(\frac{x^{t+k}_{MBB}-x^{t-i}_{MBB}}{k+i+1}\right)^2+\left(\frac{y^{t+k}_{MBB}-y^{t-i}_{MBB}}{k+i+1}\right)^2} \Big/ 2k$

Note: CFV_{bm} is the CFV for body movement, and CFV_{bt} is the CFV for body translation. *c_mb_sz*, *c_mb_ht* and *c_mb_wd* represent *change of MBB size, height and width*, respectively. h_{mbb} is the height for object's MBB, w_{mbb} is the width for object's MBB, (x_{MBB}, y_{MBB}) is the center of MBB. t is the current frame number. t-k and t+k indicate previous and future k frames relative to the current frame. In our experiment, we set k to 4.

TABLE III
K-L DISTANCE FOR FEATURES IN TABLE II FOR ONE SET OF TRAINING DATA

	$f_{c\_mb\_sz}$	$f_{c\_mb\_wd}$	$f_{c\_mb\_ht}$	f_{avg_spd}	f_{avg_vct}	f_{mean_vct}
$f_{c\_mb\_sz}$		34.42	19.35	9463.4	6471.5	6058.8
$f_{c\_mb\_wd}$	34.42		13.36	8683.1		6239.4
$f_{c\_mb\_ht}$	19.35	13.36		4773.1	3770.5	3382.1
f_{avg_spd}	9463.4	8683.1	4773.1		11.37	12.95
f_{avg_vct}	6471.5	6941.2	3770.5	11.37		11.90
f_{mean_vct}	6058.8	6239.4	3382.1	12.95	11.90	

(Circles: CFVs are formed by grouping features with small distances)

by the K-L distances in (2), with $\mathrm{CFV_{bm}}$ for the category *body movement*, and $\mathrm{CFV_{bt}}$ for the category *body translation*. The K-L distances between the features in Table II for *one set of training data* are listed in Table III. The grouping result is shown by the circles in Table III. The matrix is similar for other training sets and the grouping results are the same.

In order to exclude the effect of a tracking algorithm, we use the ground-truth tracking data which is available in the PETS'04 dataset to get the MBB information. In practice, various practical tracking methods [10], [23] can be used to obtain the MBB information. Furthermore, the features in Table II are calculated by averaging several consecutive frames to improve the robustness to the possible tracking error.

Due to the inclusion of the local model, more features for the M-DPF dissimilarity checking become useful. The new added features are listed in Table IV. When checking the M-DPF dissimilarity by (7), we set $r=2$, $k_i=1$, and $w_i=1/\sigma_i$, where σ_i is the standard deviation of feature f_i. The w_i's for the features of x^t_{MBB}, y^t_{MBB}, and d^t_{ob} are set to 1. We discard the two features with the largest distances [i.e., $n=2$ in (7)].

1) Frame-Error-Rate Comparison for Different Methods: In this experiment, we compare the frame-level error rate of the following five methods for fusing results from multiple streams. Frame-level error rate measures the recognition accuracy for *each* video frame.

TABLE IV
NEW ADDED FEATURES FOR THE DPF DISSIMILARITY CHECKING

Feature Name	**Definition**
x^t_{MBB}	x-axis position of the center of MBB
y^t_{MBB}	y-axis position of the center of MBB
d^t_{ob}	Duration: # of frames since the object *ob* appears
h_{mbb}	height for object's MBB
w_{mbb}	width for object's MBB
r^t_{mbb}	$r^t_{mbb}=h^t_{mbb}/w^t_{mbb}$
$size^t_{mbb}$	$size^t_{mbb}=h^t_{mbb}\cdot w^t_{mbb}$

1) **Weighted Average** [37], [38] (WA in Table V). Use a weighted average of results from the two CFVs, as in

$$t=A_k \quad if\ \left(\sum_i w_{i,k}\cdot P\left(A_k|F_i(t)\right)\right) > \max_{j,j\neq k}\left\{\sum_i w_{i,j}\cdot P\left(A_j|F_i(t)\right)\right\} \tag{9}$$

where t is the current frames (or sample). The definition of $w_{i,j}$ and $P(A_j|F_i(t))$ are the same as in (5).

2) **Weighted Multiplication** [35], [36] (WM in Table V). The results for the two classifiers are combined by $p(F_{bm}|A_k)^{\mathrm{wk}} \bullet p(F_{bt}|A_k)^{(1-\mathrm{wk})}$, where $p(F_{bm}|A_k)$ and $p(F_{bt}|A_k)$ are GMM distributions for $\mathrm{CFV_{bm}}$ and $\mathrm{CFV_{bt}}$. w_k is the weight representing the relative reliability of the two CFVs for A_k.

3) **AVC method** [22] (AVC in Table V). In [22], the histograms of audio and video features are combined to form an audio–video co-occurrence (AVC) matrix. In our experiment, we create two labeled histograms for the two CFVs for each activity (based on the method in [10]) and use them to replace the histograms of audio and video features in [22]. There will be one AVC matrix for each activity. After the AVC matrix for activity A_k is created, the activity A_k can then be detected based on the AVC matrix.

4) **Early Integration** [35] (EI in Table V). Use *one* GMM model for the whole six features in Table II.

5) **The proposed CFR algorithm** (CFR in Table V). Use the weighted average of GMM as a global model to detect *confident frames* and use them as the local model, and then combine the global and local models to detect the *left frames*.

The experiments are performed under 50% training and 50% testing. We perform five independent experiments and average the results. The results are shown in Table V. In order to show the contribution of each individual CFV, we also include the results of using only the $\mathrm{CFV_{bm}}$ classifier ($\mathrm{C_{bm}}$ in Table V) or only $\mathrm{CFV_{bt}}$ classifier ($\mathrm{C_{bt}}$ in Table V).

In Table V, the Misdetection (*Miss*) rate and the false alarm (FA) rate [10] are compared. In the last row of Table V, we include the total frame error rate (TFER) which is defined by

TABLE V
FRAME-LEVEL ERROR RATE RESULTS FOR 50% TRAINING AND 50% TESTING

		C_{bt}	C_{bm}	WA	WM	AVC	EI	CFR
Inactive	Miss (%)	3.3	4.8	0.61	0.43	6.43	6.9	0.87
	FA (%)	3.1	2.9	6.2	6.91	0.83	1.4	1.6
Active	Miss (%)	40.1	56.8	46.6	43.57	20.21	25.61	12.9
	FA (%)	2.72	4.06	0.51	0.76	1.64	1.87	1.09
Walking	Miss (%)	4.02	7.15	2.6	2.87	8.32	1.32	2.41
	FA (%)	11.84	19.25	16.82	16.1	3.9	5.57	5.11
Running	Miss (%)	56.7	90.31	77.85	83.3	44.72	52.06	26.25
	FA (%)	0.2	0	0.04	0.01	1.43	0.4	0.27
Fighting	Miss (%)	70.2	76.53	71.41	73.16	59.57	56.62	40.81
	FA (%)	0.07	0.16	0.02	0.03	1.47	0.38	0.2
TFER		12.42	15.73	11.28	11.46	10.24	8.81	4.09

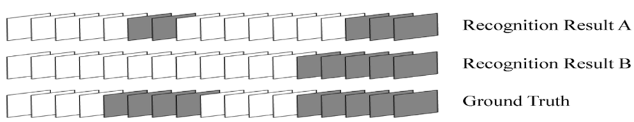


Fig. 7. Recognition results comparison. (White frame sequence: action A_1; gray frame sequence: action A_2).

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 activities.

From Table V, we can see that the proposed CFR algorithm, which introduces the local model to help detect activities, has the best recognition performance compared with other methods. Furthermore, for activities such as *Active*, *Running*, and *Fighting* where the GMM classifiers have high *Misdetection rates* (*miss* in Table V), our CFR algorithm can greatly improve the detection performance.

2) Activity-Level Error Rate Comparison: In the previous section, we showed experimental results for the frame-level error rates. However, in some scenarios, people are more interested in the error rate in the activity level (i.e., the rate of missing an activity when it happens). In these cases, frame-level error rates may not be able to measure the performance accurately. For example, in Fig. 7, the two results have the same frame-level error rates while their activity-level error rates are different (*Recognition Result A* has a lower activity-level error rate because it detects both of the A_2 actions while *Recognition Result B* only detects one).

Here, we compare the activity-level error rate performance. First, we define the time interval $[t_1, t_2]$ to be an *Activity Clip* of activity A_k if

$$\begin{cases} L_{\text{grd}}(t_1) \neq L(t_1-1), & (\text{not } A_k \text{ before } t_1) \\ L_{\text{grd}}(t) = A_k, \forall t \in [t_1, t_2], & (\text{all label is} A \text{ during } [t_1, t_2]) \\ L_{\text{grd}}(t_2) \neq L(t_2+1), & (\text{not } A_k \text{ after } t_2) \end{cases}$$

where $L_{\text{grd}}(t)$ is the ground-truth activity label of frame t.

The activity-level error rate (AER) in this experiment is then defined as AER $= N_{k,\text{miss}}/N_{k,\text{total}}$, where $N_{k,\text{miss}}$ is the total number of *missed Activity Clips* in (10) for activity A_k. $N_{k,\text{total}}$ is the total number of *Activity Clips* for A_k. An *Activity Clip* of time interval $[t_1, t_2]$ is a *missed Activity Clip* if

TABLE VI
ACTIVITY-LEVEL ERROR RATE RESULTS FOR 50% TRAINING AND 50% TESTING

	C_{bt}	C_{bm}	WA	WM	AVC	EI	CFR
Inactive	1.8%	2.04%	0.6%	0.6%	3.6%	1.2%	0.6%
Active	17.67%	20.3%	12.67%	13.33%	15.5%	12.67%	9.56%
Walking	3.77%	5.68%	3.4%	4.32%	6.16%	3.86%	2.31%
Running	41.33%	63.67%	36.33%	43%	27.33%	36.33%	24%
Fighting	50%	50%	50%	50%	46.67%	50%	36.67%

$$L_{\text{grd}}(t) \neq L_{\text{recognized}}(t), \quad \forall t \in [t_1, t_2] \tag{10}$$

where $L_{\text{grd}}(t)$ is the ground-truth activity label at frame t. $L_{\text{recognized}}(t)$ is the recognition result at frame t.

In Table VI, we compared the AER performance of the five methods described in Section VI-A1.

Some important observations from Table VI are listed below.

1) Compared with the *Frame-level Misdetection (Miss) Rates* in Table V, some methods have much closer performances in AERs (e.g., the Miss rate for *running* of EI in Table V is more than 25% lower than that of WA, however, their AERs are the same in Table VI). This is because these two rates (Miss and AER) reflect different aspects of the recognition methods. The Miss rate reflects more on the ability of the methods to precisely locate the boundary of events (i.e., the ability to recognize all frames between the starting and ending points of the events), while the AER reflects more on the ability of the methods to detect events when they happen (i.e., the ability to detect at least one frame when the event occurs).

   Comparing Table V and Table VI, we find that most methods have a much lower AER than the Miss rate (especially for activities with high Miss rates such as *active*, *running,* and *fighting*). This means that most of these methods are more capable of detecting the existence of the activities than precisely locating their boundaries. Compared with these methods, the proposed CFR algorithm has a similar AER but a greatly improved Miss rate. This shows that CFR can locate the activity boundaries more precisely.
2) The CFR uses weighted average (WA) to detect the confident frames as the local model. This means that, if an activity clip is missed by WA, CFR will also fail to detect any confident frames in the same activity clip. However, the result in Table VI shows that many of the AERs of CFR are lower than those of WA. This is because, when WA fails to detect any confident frame during an activity clip of A_k, CFR may still be able to detect the event by checking the dissimilarity with local models of A_k outside the clip.
3) Based on the previous two observations, we see that the introduction of the local model in CFR has two effects:
   a) it helps detect the left frames within its own activity clip, thus locating the clip boundary more precisely and also reducing the frame-level error rates (Miss and FA);

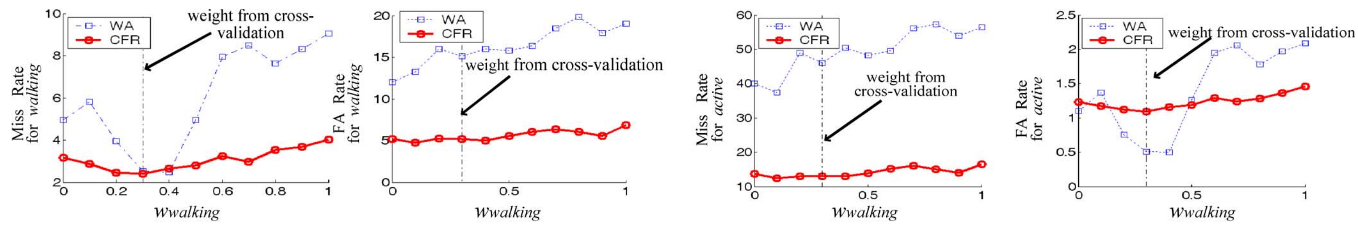


Fig. 8. Comparison of the impact of changing w_k to the recognition performances for WA and CFR. (a) The impact of w_{walking} to the performance of walking (Left: Miss; right: FA). (b) The impact of w_{walking} to the performance of active (Left: Miss; right: FA).

b) it helps detect other activity clips where no confident frame is detected, thus reducing the misdetection rate for activity clips.

4) From Table VI, we can see that the AERs of CFR for most activities are close to those of WA. This means that the AER performance of CFR mainly depends on the algorithm to detect confident frames. Therefore, a suitable confident-frame detection method is important. In this paper, WA is used for detecting confident frames. However, other methods such as WM and AVC can also be applied if they have better performance.

3) Experimental Results for Weights and Thresholds Selection: In several methods such as WA and WM, we need to select a suitable weight [i.e., the $w_{i,k}$ in (9)] to fuse the results from two CFVs. Furthermore, since the CFR algorithm uses WA to detect the confident frames, the weights and thresholds [i.e., $w_{i,k}$ and $th_{i,k}$ in (5)] also need to be selected for confident frame detection. In the previous experiments, all these weights and thresholds are selected by the fivefold cross-validation method [35]. However, the cross-validation is relatively complicated. We need to try all the possible combinations of parameters. Furthermore, the complexity of the cross-validation algorithm will increase exponentially with the increasing number of parameters. As mentioned in Section V-C, our proposed CFR algorithm is more robust to the change of weight values since the weights only need to work well on confident frames rather than the whole testing data. This implies that with the proposed CFR algorithm, we may be able to use a rough weight or use a simpler way to select the parameters. In the following, we show two experimental results to justify this claim.

a) Experiment 1 for Parameter Selection: In this experiment, we justify our claim that the recognition performance of our CFR algorithm is robust to the change of parameters. Since the CFR in this paper uses the same method as WA to detect confident frames, we will focus on the comparison of these two methods. Furthermore, since we only have two CFVs in the experiment, the CFR and WA algorithm in (5) and (9) can be rewritten, respectively, as

$$\begin{aligned} \text{CFR}: t = T_k,\ \text{if}\ & (w_k \cdot P(A_k|F_{bm}) \\ & +(1-w_k)\cdot P(A_k|F_{bt})) > th_k \qquad (11) \\ \text{WA}: t = A_k,\ \text{if}\ & (w_k \cdot P(A_k|F_{bm}) + (1-w_k)\cdot P(A_k|F_{bt}) \\ & > \arg \max_{A_i, A_i \neq A_k} \{w_i \cdot P(A_i|F_{bm}) \\ & +(1-w_i)\cdot P(A_i|F_{bt})\} \qquad (12) \end{aligned}$$

where F_{bm} and F_{bt} represent the features for CFV_{bm} and CFV_{bt}, respectively. The definition of w_k, t, A_k, T_k, and th_k are the same as in (5) and (6).

We first select the parameters [w_k and th_k in (11) and (12)] by cross validation. The parameter values selected from the validation set is defined as w_k^{Valid} and th_k^{Valid}. Then, we change the weight value w_k for one activity A_k and keep the weight value for other activities unchanged. For the CFR algorithm, we also keep the threshold th_k for *all* activities unchanged. We then use the changed parameter set to perform recognition on the testing data and plot the recognition performance changes.

Fig. 8(a) and (b) shows the recognition performance (Miss and FA) change for activities under different w_k values of activity $walking$ (i.e., $w_k = w_{\text{walking}}$). It is the result from one experiment of 50% Training and 50% Testing. The results from other experiments are similar. Fig. 8(a) shows the impact of changing w_{walking} to the recognition performance for walking, and Fig. 8(b) shows the impact of w_{walking} to active. Fig. 8 shows results when w_{walking} is changed. Similar observations can be found when the weights of other activities are changed.

From Fig. 8(a) and (b), we can see that the performance of the WA method fluctuates substantially with the change of w_k. This reflects that the recognition performance of WA is very sensitive to the change of w_k. On the contrary, the recognition performances of our CFR algorithm are quite stable with the change of w_k. The performance of CFR is close to those under w_k^{Valid} even when w_k is far from w_k^{Valid} (the dashed vertical line). This justifies that CFR is *robust* to the change of w_k.

Since CFR also uses threshold th_k to detect confident frames, a similar experiment is performed on th_k to see its impact on the recognition performances of CFR. We fix all w_k's to be w_k^{Valid}. Then we change the value of th_k for activity A_k and keep the threshold value for other activities unchanged. The recognition performances under different th_k values of walking (i.e., $th_k = th_{\text{walking}}$) are plotted in Fig. 9(a) and (b). Fig. 9(a) shows the impact of changing $th_{walking}$ to the performance of $walking$. Fig. 9(b) shows the impact of th_{walking} to the performance of *active*. Three observations from Fig. 9 are listed below.

1) The performance of CFR is stable when th_k changes within a certain range around th_k^{Valid} (the vertical lines in Fig. 9). This means that *CFR is also robust to the change of th_k within a certain range around th_k^{Valid}*. We call this range *stable range*.

2) *A too small or too large value of the threshold th_k will obviously decrease the recognition performance of CFR*. A too small threshold value may include many *false alarm*

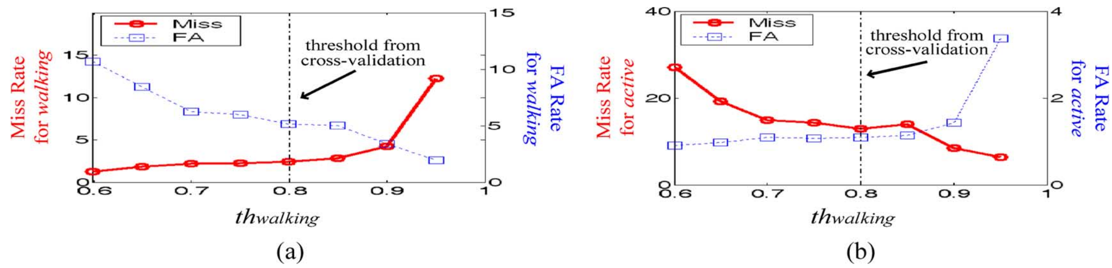


Fig. 9. Comparison of the impact of changing th_{walking} to the recognition performances of CFR. (a) Performance change for *walking* under different th_{walking}. (b) Performance change for *active* under different th_{walking}.

TABLE VII
RESULTS UNDER ROUGHLY SELECTED PARAMETERS

		WA_{C-V}	WA_R	WA_{C-V}	WM_R	CFR_{C-V}	CFR_R
Inact-ive	Miss	0.61%	3.82%	0.43%	2.58%	0.87%	0.94%
	FA	6.2%	4.71%	6.91%	5.16%	1.6%	1.76%
Act-ive	Miss	46.6%	57.24%	43.57%	50.03%	12.9%	15.23%
	FA	0.51%	2.25%	0.76%	3.41%	1.09%	1.57%
Walk-ing	Miss	2.6%	5.29%	2.87%	6.1%	2.41%	2.17%
	FA	16.82%	19.3%	16.1%	18.22%	5.11%	5.69%
Run-ning	Miss	77.85%	88.11%	83.3%	92.53%	26.25%	30.08%
	FA	0.04%	0.17%	0.01%	0.11%	0.27%	0.23%
Fight-ing	Miss	71.41%	82.59%	73.16%	80.4%	40.81%	45.1%
	FA	0.02%	0.15%	0.03%	0.22%	0.2%	0.19%
TFER		11.28%	15.48%	11.46%	15.53%	4.09%	4.62%

(The gray columns named "$C-V$" are results under cross-validation parameters; the white columns named "R" are results under roughly set parameters.)

samples as confident frames (an extreme case: if $th_k = 0$, it will be exactly the same as the WA method). On the other hand, a too large threshold value may reject most of the samples, making the recognition difficult (an extreme case: if $th_k = 1$, there will be almost no *confident frames* detected).

3) Different activities A_k may have different th_k^{Valid}. However, since each th_k has a *stable range* around th_k^{Valid}, we may still be able to find a *common stable range* for all activities. Our experiments imply that values between 0.65 and 0.8 may be a suitable choice of thresholds for most activities.

The results from Figs. 8 and 9 justify that our CFR algorithm is robust to the change of parameters w_k and th_k. This advantage implies that for the CFR algorithm, we may be able to set the parameters to rough specific values or by a simplified parameter-selection method such as increasing the searching step-size or decreasing the searching range, instead of using the complicated cross-validation method to select the parameters. This is further justified in the following experiment.

b) Experiment 2 for parameter selection: In this experiment, we set the w_k for all activities to be 0.5 and th_k for all activities to be 0.7 and then use this parameter set to recognize the activities. We perform five experiments with 50% training and 50% testing and average the result (the same setting as Table V). The experimental results are listed in Table VII. In order to compare with the results under *cross-validation* parameters, we attach the results of Table V (the gray columns).

In Table VII, three methods are compared (WA, WM, and CFR). From Table VII, we can see that our proposed CFR algorithm can still perform well under the roughly selected parameters while the performances of both WA and WM methods decrease significantly under this situation. This validates that the CFR algorithm allows us to select parameters through more simplified methods with small impact on the performance. As we will see in Section VI-C, this advantage also increases the flexibility of our algorithm for adding new events.

TABLE VIII
PROPOSED METHOD IN DEALING WITH INSUFFICIENT DATA

		C_{bt_Dt}	C_{bt_Adt}	C_{bm_Dt}	C_{bm_Adt}	CFR_{Dt}	CFR_{Adt}
Run-ning	Miss	56.7%	37.86%	90.31%	70.72%	26.25%	14.32%
	FA	0.2%	0.21%	0%	0.28%	0.27%	0.32%
Fight-ing	Miss	70.2%	64.52%	76.53%	68.3%	40.81%	29.61%
	FA	0.07%	0.1%	0.16%	0.15%	0.2%	0.22%

(The gray columns labeled as "Dt" are results whose models are modeled directly from the training data; the white columns labeled as "Adt" are results whose models are adapted by our proposed method.)

B. Experimental Results for Training LTS Events

From Table V, we can see that the misdetection rate (Miss) for activities such *Running* and *Fighting* are relatively high (although our CFR algorithm has significantly improved the *mis*detection rate from other methods). This is because the number of training samples in Table I is small. The training samples are not sufficient to model the whole distribution of these activities, which reduces the prediction capability of these models for the unknown data.

We use our proposed LTS event training method to deal with the insufficient training data problem, where we adapt both CFVs' GMM models of *Running* from *Walking*, while both CFV GMM models of *Fighting* are adapted from *Active* (which is different from Fig. 3 because *running* itself is also lacking training data). The recognition results based on our adapted-GMM models are shown in Table VIII.

The results in Table VIII show the effectiveness of our proposed method in dealing with insufficient training data. We can see that although improved by our algorithm, the misdetection rate for *fighting* is still relatively high. The main reason for this is that, besides lacking training data, the features we use (in Table II) are relatively simple (all from MBB), while the feature distributions of these activities are similar to other activities, making the classification difficult. In order to improve the performance further for the activities, more sophisticated features can be used, or the interaction between different objects can be considered, which will be our future work.

C. Experiment Results for the Flexibility of Adding New Events

We give an example to illustrate the flexibility of adding a new event. In this example, a CFV-based system with two CFVs defined by Table II has been trained to detect five activities: *Inactive*, *Active*, *Walking*, *Running*, and *Fighting*. We define a **new event** "*picking up or leaving a bag*." Since there is no ground-truth label for *picking up or leaving a bag* in the dataset, we label it manually. The total number of positive samples for "*picking up or leaving a bag*" is 366. Note that these samples have been excluded from the dataset in the previous experiments so that they are *new* to the system when the event is added.

As mentioned in Section III, when new events are added to the system, the existing CFVs may not be enough to differentiate all activities, necessitating the adding of new CFVs. In this example, we *assume* that the two existing CFVs in Table II are not enough for differentiating the new "*picking up or leaving a bag*" event. Therefore, we add one more CFV named $\mathrm{CFV}_{\mathrm{Change\_of\_Body\_Ratio}}$. In $\mathrm{CFV}_{\mathrm{Change\_of\_Body\_Ratio}}$, there is only one feature $f_{c\_\mathrm{mbb\_ratio}}$ which represents the change of MBB ratio. The new CFV is defined as

$$\begin{cases} \mathrm{CFV}_{\mathrm{Change\_of\_Body\_Ratio}} = [f_{c\_\mathrm{mbb\_ratio}}] \\ f_{c\_\mathrm{mbb\_ratio}} = \sum_{i=t-k}^{t+k} \left| \dfrac{\dfrac{w^i_{\mathrm{mmb}}}{h^i_{\mathrm{mmb}}} - \dfrac{w^{i-1}_{\mathrm{mmb}}}{h^{i-1}_{\mathrm{mmb}}}}{\frac{w^{i-1}_{\mathrm{mmb}}}{h^{i-1}_{\mathrm{mmb}}}} \right| \Big/ 2k+1 \end{cases}$$

where k, w^i_{mmb}, and h^i_{mmb} are the same as in Table II.

Then, the flexibility of our algorithm for adding new events in this example can be described in the following two points.

1) When the new event *picking up or leaving a bag* was added to the system, we do not need to change or retrain the $\mathrm{CFV}_{\mathrm{bm}}$ and $\mathrm{CFV}_{\mathrm{bt}}$ models for events *inactive*, *active*, *walking*, *running*, and *fighting*. We only need to train the $\mathrm{CFV}_{\mathrm{Change\_of\_Body\_Ratio}}$ models for these events as well as all the three CFV models for the new event *picking up or leaving a bag*. The models that need training (white circles) and models that do not need training (gray circles) in this example are shown in Fig. 10. In practical situations, the number of models that do not need training is much larger than the number of models that need training.
2) Since the new $\mathrm{CFV}_{\mathrm{Change\_of\_Body\_Ratio}}$ is added for each event, we need to update parameters that fuse these CFV models [$w_{i,k}$ and th_k in (5))] However, as mentioned, the CFR algorithm is *robust* to the change of these parameters. This means that we can set these parameters roughly or by a simple parameter selection method, instead of performing the complicated cross-validation method to update the parameters.

Based on the above two points, in the experiment, we can train the new system through a simple way by: 1) only training the white labeled CFV models in Fig. 10 and 2) setting the weights and thresholds roughly (here, we set all weights to be 0.33 and all thresholds to be 0.7). Table IX (white column) shows the results for 50% training and 50% testing (the setting is the same as in Table V). Table IX (gray column) shows the recognition results under cross-validation parameters.

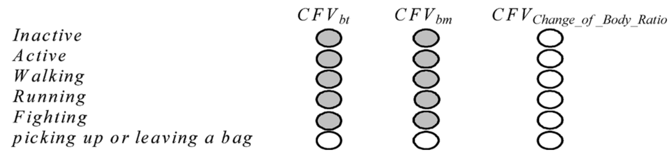


Fig. 10. Models need training or do not need training. (Gray circles: models do not need training; white circles: models need training.)

TABLE IX
EXPERIMENTAL RESULTS FOR ADDING NEW EVENT

		$\mathrm{CFR}_{\mathrm{C\text{-}V}}$	$\mathrm{CFR}_{\mathrm{R}}$
Inactive	Miss	0.9%	0.95%
	FA	1.43%	1.88%
Active	Miss	16.26%	18.52%
	FA	1.02%	1.6%
Walking	Miss	3.07%	2.86%
	FA	5.2%	5.96%
Running	Miss	23.94%	27.8%
	FA	0.31%	0.28%
Fighting	Miss	46.31%	51.55%
	FA	0.16%	0.11%
Picking up or leaving a bag	Miss	36.94%	40.5%
	FA	0.19%	0.17%
TFER		4.55%	5.12%

(The gray columns named "$C - V$" are results under cross-validation parameters; the white columns named "R" are results under roughly set parameters)

From Table IX, we can see that, when the system is adapted to include the new event through a simple manner by our algorithm, we still can achieve good results close to those under cross-validation parameters. This justifies the flexibility of the algorithm.

VII. CONCLUSION

In this paper, we made the following three contributions. First, we proposed to represent activities by the combination of CFV-based models which has good flexibility in representing activities as well as in handling new events. Second, based on the CFV-based representation, we proposed a method to deal with the model training problems for events which lack training data (LTS events). Finally, we also proposed a confident-frame-based recognition algorithm which is capable of improving the recognition accuracy. Experimental results demonstrate the effectiveness of our proposed methods.

REFERENCES

[1] P. Harmo, T. Taipalus, J. Knuuttila, J. Wallet, and A. Halme, "Needs and solutions—Home automation and service robots for the elderly and disabled," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 2005, pp. 3201–3206.

[2] D. Chen, J. Yang, and H. Wactlar, "Towards automatic analysis of social interaction patterns in a nursing home environment from video," in *Proc. ACM Int. Workshop Multimedia Inf. Retrieval*, 2004, pp. 283–290.

[3] D. Hall, L. Eubanks, L. S. Meyyazhagan, R. D. Denny, and S. C. Johnson, "Evaluation of covert video surveillance in the diagnosis of Munchausen syndrome by proxy: Lessons from 41 cases," *Pediatrics*, vol. 105, no. 6, pp. 1305–1312, Jun. 2000.

[4] D. Ayers and M. Shah, "Monitoring human behavior from video taken in an office environment," *Image Vis. Computing*, vol. 19, pp. 833–846, 2001.

[5] L. Zelnik-Manor and M. Irani, "Event-based video analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2001, vol. 2, pp. 123–130.

[6] H. Zhong, J. Shi, and M. Visontai, "Detecting unusual activity in video," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2004, vol. 2, pp. 819–826.

[7] D. Zhang, D. Gatica-Perez, S. Bengio, and I. McCowan, "Semi-supervised adapted HMMs for unusual event detection," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2005, vol. 1, pp. 611–618.

[8] A. Datta, M. Shah, and N. Lobo, "Person-on-person violence detection in video data," in *Proc. Int. Conf. Pattern Recognit.*, 2002, vol. 1, pp. 433–438.

[9] J. Snoek, J. Hoey, L. Stewart, and R. S. Zemel, "Automated detection of unusual events on stairs," in *Proc. Can. Conf. Comput. Robot Vis.*, 2006, pp. 5–12.

[10] 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 Eval. Tracking and Surveillance*, 2004, pp. 33–40.

[11] P. C. Ribeiro and J. Santos-Victor, "Human activity recognition from video: Modeling, feature selection and classification architecture," in *Proc. Int. Workshop Human Activity Recognition and Modeling*, 2005, pp. 61–70.

[12] T. V. Duong, H. H. Bui, D. Q. Phung, and S. Venkatsh, "Activity recognition and abnormality detection with the switching hidden semi-Markov model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2005, vol. 1, pp. 838–845.

[13] S. Gong and T. Xiang, "Recognition of group activities using dynamic probabilistic networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2003, vol. 2, pp. 472–479.

[14] N. Oliver, E. Horvitz, and A. Garg, "Layered representations for human activity recognition," in *Proc. IEEE Conf. Multimodel Interfaces*, 2002, pp. 3–8.

[15] C. Liu, P. Chung, and Y. Chung, "Human home behavior interpretation from video stream," in *Proc. IEEE Int. Conf. Networking, Sensing and Control*, 2004, vol. 1, pp. 192–197.

[16] P. Viola, M. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," *Int. J. Comput. Vis.*, vol. 63, no. 2, pp. 153–161, Jul. 2005.

[17] T. Chaodhury, J. Rehg, V. Pavlovic, and A. Pentland, "Boosted learning in dynamic bayesian networks for multimodal detection," in *Proc. Int. Conf. Inf. Fusion*, 2002, vol. 1, pp. 550–556.

[18] 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.

[19] 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.

[20] CAVIAR Project [Online]. Available: http://www.homepages.inf.ac.uk/brf/CAVIAR/

[21] B. Li, E. Chang, and C. T. Wu, "DPF-a perceptual distance function for image retrieval," in *Proc. Int. Conf. Image Process.*, 2002, vol. 2, pp. 597–600.

[22] M. Cristani, M. Bicego, and V. Murino, "Audio-visual event recognition in surveillance video sequences," *IEEE Trans. Multimedia*, vol. 9, no. 2, pp. 257–267, Feb. 2007.

[23] 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.

[24] A. A. Efros, A. C. Berg, G. Mori, and J. Malik, "Recognizing action at a distance," in *Proc. IEEE Conf. Comput. Vis.*, 2003, vol. 2, pp. 726–733.

[25] D. Koller and M. Sahami, "Toward optimal feature selection," in *Proc. 13th Conf. Machine Learning*, 1996, pp. 284–292.

[26] S. Amari and S. Wu, "Improving support vector machine classifiers by modifying kernel functions," *Neural Networks*, vol. 12, pp. 783–789, 1999.

[27] G. Wu and E. Y. Chang, "Adaptive feature-space conformal transformation for imbalanced-data learning," in *Proc. 12th Int. Conf. Machine Learning*, 2003, pp. 816–823.

[28] J. P. Hoffbeck and D. A. Landgrebe, "Covariance matrix estimation and classification with limited training data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 6, pp. 763–767, Jun. 1996.

[29] C. E. Thomaz, D. F. Gillies, and R. Q. Feitosa, "A new covariance estimate for Bayesian classifiers in biometric recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 2, pp. 214–223, Feb. 2004.

[30] P. Mitra, C. A. Murthy, and S. K. Pal, "Unsupervised feature selection using feature similarity," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 3, pp. 301–312, Mar. 2002.

[31] B. Moghaddam and A. Pentland, "Probabilistic visual learning for object representation," *IEEE Trans. Pattern Anal. Mach. Intel.*, vol. 19, no. 7, pp. 696–710, Jul. 1997.

[32] F. I. Bashir, A. A. Khokhar, and D. Schonfeld, "Real-time motion trajectory-based indexing and retrieval of video sequences," *IEEE Trans. Multimedia*, vol. 9, no. 1, pp. 58–65, Jan. 2007.

[33] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digit. Signal Process.*, vol. 10, pp. 19–41, 2000.

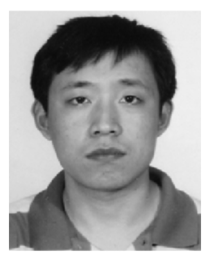
[34] J. Bilmes, A Gentle Tutorial of the EM Algirthm and Its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models U.C. Berkeley, ICSI-TR-97-021, 1997.

[35] 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.

[36] S. Dupont and J. Luettin, "Audio-visual speech modeling for continuous speech recognition," *IEEE Trans. Multimedia*, vol. 2, no. 3, pp. 141–151, Sep. 2000.

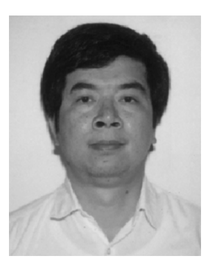
[37] I. Bloch, "Information combination operators for data fusion: A comparative review with classification," *IEEE Trans. Syst., Man Cybern.*, vol. 26, no. 1, pp. 52–67, Jan. 1996.

[38] J. Tang, J. Gu, and Z. Cai, "Data fusion with different accuracy," in *Proc. IEEE Conf. Robot. Biomimet.*, 2004, pp. 811–815.



Weiyao Lin (S'08) received the B.E. and M.S. degrees from Shanghai Jiao Tong University, Shanghai, China, in 2003 and 2005, respectively, both in electrical engineering. He is currently working toward the Ph.D. degree in electrical engineering from the University of Washington, Seattle.

His 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, in 1976, the M.S. degree from the University of Texas at Arlington in 1981, and the Ph.D. degree from University of California, Los Angeles, in 1985, all in electrical engineering.

He joined the University of Washington, Seattle, in August 1996, where he is currently a Professor. Previously, he was the Director of the Video Signal Processing Research Group at Bellcore. He was a chaired Professor with TsingHwa University, Beijing, China, and a Visiting Professor with Tokyo University 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 coedited a book, *Compressed Video over Networks* (Marcel Dekker, 2000).

Dr. Sun was the Editor-in-Chief of the IEEE TRANSACTIONS ON MULTIMEDIA and a Distinguished Lecturer of the IEEE Circuits and Systems Society (CAS-S) from 2000 to 2001. He was the recipient of the IEEE CAS-S Golden Jubilee Medal in 2000, the IEEE TCSVT Best Paper Award in 1993, and the Award of Excellence from Bellcore for his work on the digital subscriber line in 1987. He 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 (TCVST) from 1995 to 1997. From 1988 to 1991, he was the chairman of the IEEE CAS Standards Committee and established the IEEE Inverse Discrete Cosine Transform Standard.

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 Lab (NSL), Electrical Engineering Department, University of Washington, Seattle. His research interests are in the areas of applied cryptography for multiuser environment, wireless networking, and applications of information theory to security. He is a coeditor of the book *Secure Localization and Time Synchronization in Wireless Ad Hoc and Sensor Networks* (Springer-Verlag, 2007).

Dr. Poovendran was a recipient of the NSA Rising Star Award and Faculty Early Career Awards, including the National Science Foundation CAREER Award in 2001, the Army Research Office YIP Award in 2002, the Office of Naval Research YIP Award in 2004, PECASE in 2005 for his research contributions to multiuser security, and a 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.

Zhengyou Zhang (SM'97–F'05) received the B.S. degree in electronic engineering from the University of Zhejiang, Hangzhou, 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, and the Ph.D. degree in computer science (specialized in computer vision) and the Doctor of Science (Habilitation a diriger des recherches) diploma from the University of Paris XI, Paris, France, in 1990 and 1994, respectively.

He is a Principal Researcher with Microsoft Research, 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 March 1998. In 1996–1997, he spent a one-year sabbatical as an Invited Researcher with the Advanced Telecommunications Research Institute International (ATR), 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* (Springer-Verlag, 1992); *Epipolar Geometry in Stereo, Motion and Object Recognition* (Kluwer, 1996); and *Computer Vision* (Chinese Academy of Sciences, 1998, in Chinese). He has given a number of keynotes in international conferences.

Dr. Zhang is an Associate Editor of the IEEE TRANSACTIONS ON MULTIMEDIA, an Associate Editor of the *International Journal of Computer Vision*, an Associate Editor of the *International Journal of Pattern Recognition and Artificial Intelligence*, 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, 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 (ICCV2003), October 2003, Nice, France, a Program Co-Chair of the Asian Conference on Computer Vision (ACCV2004), January 2004, Jeju Island, Korea, a Demo Chair of the International Conference on Computer Vision (ICCV2005), October 2005, Beijing, China, a Program Co-Chair of the International Workshop of Multimedia Signal Processing (MMSP), Victoria, BC, Canada, October 2006, and a Program Co-Chair of International Workshop on Motion and Video Computing, November 2006, Austin, TX.