Multiview Activity Recognition in Smart Homes with Spatio-Temporal Features

Chen Wu, Amir Hossein Khalili and Hamid Aghajan
Stanford University, Stanford CA

***Abstract*—Recognizing activities in a home environment is challenging due to the variety of activities that can be performed at home and the complexity of the environment. Multiple cameras are usually needed to cover the whole observation area. This adds camera fusion as another challenge to activity recognition. We propose a hierarchical approach that recognizes both coarse-level and fine-level activities, in which different image features and learning methods are used for different activities based on their characteristics. The paper focuses on discussing the second-level of activity recognition with spatio-temporal features. Specifically, three fusion approaches for multiview activity recognition with spatio-temporal features are presented, including two decision fusion methods and one feature fusion method. They are comparatively analyzed in terms of their tradeoffs on assumptions on system setup, model transferability and recognition rate. Experiments show that challenging activities with subtle motions such as eating, cutting, scrambling, typing, reading etc. can be recognized with our approaches.**

I. Introduction

Recognizing activities especially fine-level activities in a home environment is challenging. Therefore, different types of sensors have been used to sense user's activities in smart environments. Examples include state-change sensors [1] attached to appliances and RFID tags and readers used with household items [2] to collect object usage data as an indirect way to infer user activity. Logan et al. in [3] study activity recognition with a variety of sensors including RFID sensors, switch sensors, and motion sensors, and offer an evaluation in real-world conditions. They show that with the named sensors it is difficult to detect fine-level activities such as "reading" and "eating". They also state that visual sensing provides more information which is oftentimes complementary to other sensors.

Vision-based human activity analysis has seen significant progress in recent years [4], including advances in analyzing realistic activities from videos of the public domain [5]. However, there are only a few works that focus on activity recognition in the home environment. In [6], situation models are extracted from video sequences of a smart environment, which are essentially semantic-level activities including both individual activities and two-person interactions. Both [7] and [8] use video data and RFID for activity recognition. Wu et al. in [7] use RFID readings and object detection from video to jointly estimate activity and object use. The learning process is bootstrapped with commonsense knowledge mined from the internet, and the object model from the video is automatically acquired without labeling by leveraging RFID readings. Their work infers activity indirectly from object use. Park et al. compare activity recognition with RFID and vision [8]. They conclude that for kitchen activities which involve more object usage and for which visual features (e.g., silhouettes) are not very distinguishable, RFID-based recognition has higher performance while vision-based recognition accuracy is higher for living room activities.

The major challenges for activity recognition in the home environment include: 1. The person is often occluded by furniture; 2. Since the person freely moves and turns around while the cameras are static, the cameras may not always have a good viewpoint to observe the activity; 3. Activities in the home can have quite disparate characteristics. While activities such as lying can be distinguished from the pose, the kitchen activities usually have simple poses with subtle hand motions; 4. A fusion mechanism is needed either at the feature or the decision level.

Considering the above-mentioned challenges and aiming at both coarse- and fine-level activity recognition, we design a hierarchical vision-based activity recognition approach for smart home applications. This approach is based on the observation that due to the different characteristics of activities, different image features and models need to be applied to recognize them. In the first level, activities are classified to standing, sitting, lying from pose-related features. In the second level, other features including spatio-temporal features, face detector and moving speed are used to further classify the activities in fine granularity. In this paper, we focus on the second-level of activity recognition which uses spatio-temporal features and supervised learning. The activities targeted in our work are challenging ones in that they involve subtle motions, such as cutting, scrambling, eating, reading books, and typing on the computer. They are also interleaved with random or transition activities which are all classified as *others*. Our experiments demonstrate that the spatio-temporal features are able to capture such motion characteristics. Another emphasis of this work is to compare three fusion methods for activity recognition with multiple cameras. As mentioned above, fusion from camera views is an indispensable part of algorithm design for a camera network. We describe two decision fusion and one feature fusion methods for spatio-temporal feature-based activity recognition, and compare their system requirement/assumption, complexity, model generality and performance.

The rest of the paper is organized as follows. Related work on vision-based activity recognition is summarized in Sec. II. Sec. III describes the overall hierarchical activity recogni-

tion system in our testbed smart home enivironment, providing the big picture of second-level activity recognition with spatio-temporal features. The three fusion methods for spatio-temporal feature-based activity recognition are described in Sec. III-B. Experiments and result analysis are presented in Sec. IV. Finally, Sec. V concludes the paper.

II. Related work

Existing research on activity recognition falls into two categories in terms of the assumptions on the number of camera views during classification stage. In the first case only one viewpoint is given while in the second case synchronized multiview sequences are available for activity recognition. In the single-view case, there are three main types of approaches. In the first type of approach, the model obtained from the training process captures the 3D pose information, then the classification estimates both the viewpoint and the activity [9], [10], [11], [12]. In [9], Lv et al. use Action Net to represent transitions between key poses in actions and viewpoint changes. Silhouettes of all key poses from selected viewpoints are generated from a synthetic figure. The best matching series of actions are tracked via Viterbi algorithm based on silhouette matching. Weinland et al. [10] developed similar work where 3D exemplars are constructed from multiple cameras during training. Given the viewpoint, the 3D exemplars are used to produce 2D projections to compare with observations, and HMM is used to represent both viewpoint transition and activity dynamics. Silhouettes generated from different views are also used in [11]. Optical flow and activity duration information are integrated as well in a conditional random field model for classification. The authors showed that having multiple features increases recognition accuracy. Yan et al. in [12] build a 4D action feature model (4D-AFM) which encodes shape and motion of the activity from multiple views. During recognition, action features are matched pairwise to features on the 4D-AFM of each action. The action with the biggest matching score is selected. The second type of approach learns the model representation w.r.t. the viewpoint [13]. Souvenir et al. use R transform (two-dimensional Radon transform) to turn 2D silhouettes into 1D signals. Each action forms an R transform surface. Isomap is then used to learn the viewpoint manifold of the R transform surfaces. Farhadi et al. [14] proposed an approach with transfer learning. Given the corresponding examples between views, they learn a model describing how the appearance features space changes with viewpoints. The learning process only needs corresponding samples between two views but does not need labeled data in the second view.

When multiple views are available, they provide more information for activity recognition, but effective fusion methods between cameras need to be developed. Feature-level fusion is used in [15], [8], [6]. The camera network setup is almost circular and symmetric in [15]. During training, the actors face a particular camera. Features from all cameras are sequentially concatenated to form a single feature vector and train the model. During test, since the orientation of the action is unknown, all orientations are tested by circularly shifting features from all cameras before combining them. The best orientation is chosen with the minimal distance between the combined feature and the model. In [6], a 3D tracker generates a 3D volume from silhouettes from multiple cameras, and then the properties of the 3D volume including covariance, speed etc. are used in activity classification. The work in [8] creates scene statistics by concatenating histogram vectors from silhouette/motion mosaics of four cameras. Decision-level fusion is briefly mentioned in [12] in multiple view recognition experiments. For each activity, scores from all views are added up. The activity with the highest score is then chosen as the recognition result. The authors point out that average recognition rate of some activities is very high when combining multiple views, which may be because that action features for these activities are very discriminative.

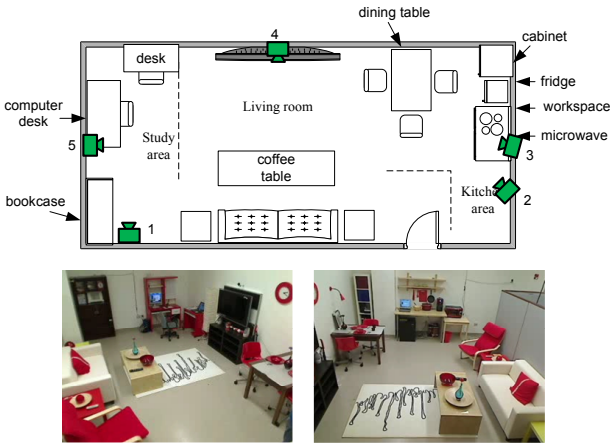

Fig. 1. The schematic and two views of AIR lab.

Spatio-temporal features have been demonstrated successful to represent activities with motion patterns. Discriminative learning methods such as SVM and nearest neighbors have been used in [16], [17]. Generative learning methods such as latent topic models have been tested on several datasets with different activities in [18].

III. Activity recognition with multiple views in the home environment

Our test-bed smart home environment, called the AIR (Ambient Intelligent Research) Lab, is a smart studio located at Stanford University (Fig. 1). It consists of a living room, kitchen, dining area, and study area. The testbed is equipped with a network of cameras, a large-screen TV, a digital window (projected wall), handheld PDA devices, appliances, and wireless controllers for lights and ambient colors.

A. The overall hierarchical activity recognition system

As the whole activity recognition system, we use a hierarchical approach to classify user activities with visual analysis in a two-level process. Different types of activities are often represented by different image features, hence attempting to classify all activities with a single approach would be

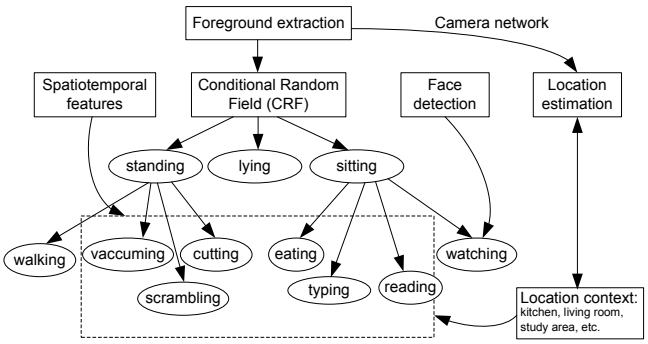

Fig. 2. Hierarchical activity analysis.

location	activity
kitchen	cutting, scrambling, vacuuming, others
dining table	eating, vacuuming, others
living room	watching, reading, vacuuming, others
study room	typing, reading, vacuuming, others

TABLE I
ACTIVITY CLASSES OF THE SECOND LEVEL AND THEIR SEMANTIC LOCATION CONTEXT.

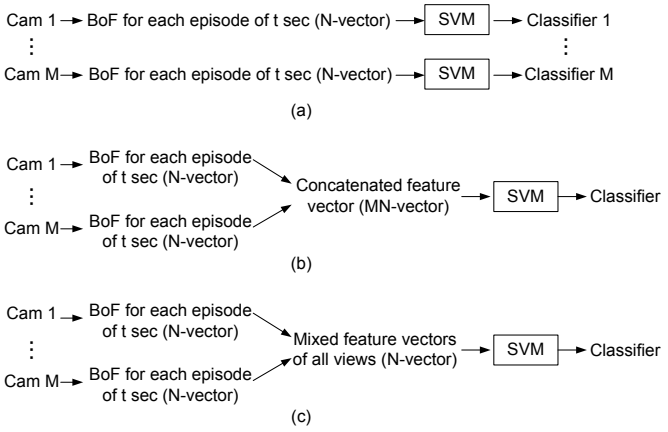

Fig. 3. Comparison of the three fusion methods. (a) best-view fusion, (b) combined-view fusion, (c) mixed-view fusion. BoF refers to bag-of-features.

ineffective. In Fig. 2, activities are represented by coarse and fine levels. The coarse activity level includes the classes of *standing*, *sitting* and *lying*, which relate to the pose of the user. Adding global motion information and face detection, more attributes are added to *standing* and *sitting* to discriminate *walking* and *watching* in the second level. The fine activity level also consists of activities involving movement such as *cutting*, *eating*, *reading*, etc. We apply such a hierarchical approach because the first-level activities are discriminated based on pose, while the second-level activities are classified based on motion features.

In the first level, activity is coarsely classified into *standing*, *sitting* and *lying* with temporal conditional random field (CRF), through employing a set of features consisting of the height of the user (through 3D tracking) and the aspect ratio of the user's bounding box. Details of the process and performance evaluation can be found in [19].

Based on the result of the coarse level, the activity is further classified at the fine-level based on spatio-temporal features [20]. A codebook of size N is constructed with K-means clustering on a random subset of all the extracted spatio-temporal features of the training dataset. Each feature is assigned to the closest cluster in Euclidean distance. The video sequences are segmented into episodes with duration of t seconds. Bag-of-features (BoF) are collected for every episode, therefore each episode has the histogram of spatio-temporal features as its feature vector. We use discriminative learning with SVM. The activities in the second level and their semantic location contexts are shown in Table I. Note that we have *others* as an activity category. This is because our sequences are not specifically designed for the defined activity types. There are many observations where the activities are in transition phase or the person is simply doing some activities at random which are not within our defined categories. This is also a challenge for our activity recognition algorithm, since due to the fact that *others* includes many different motions, the feature space for *others* is complex. However, the applications built on top of activity analysis discussed in this paper are less sensitive to false positives on *others*, because the system is usually designed to perform no operation when the user's activity is not specific.

B. Multiview activity recognition with spatio-temporal features

In this section, we describe the three fusion methods for the second level activity recognition of Fig. 2: best-view fusion, combined-view fusion and mixed-view fusion (see Fig. 3). Best-view and mixed-view fusion belong to decision-level fusion, while combined-view fusion belongs to feature-level fusion. Comparison of the fusion process is shown in Fig. 3.

Our purpose of presenting these three fusion methods is not to argue one against the others. Any of them can be used in a smart home system for activity recognition. As will be discussed later in this and the experiment sections, they have tradeoffs in assumptions on the system setup, model transferability and recognition rate. None of them is dominant enough to eliminate others. Therefore, we present the comparative analysis and performance on the three.

*1) **Best-view fusion**:* The training process is done independently for each camera, and each camera has its own activity classifier (Fig. 3(a)). The set of activities for each camera differs depending on what activities are observable in its field of view. In our experiments, since the cameras' deployment and the room layout do not change, the viewpoint of activities for each camera does not vary much. During recognition phase, the fusion process chooses the best-view camera. For each episode of t seconds, the best-view camera m is chosen and the activity classification result of camera m is determined as the person's activity. There can be different criteria for selecting the best-view camera. Our criterion here is to select the camera with the most spatio-temporal features detected in the episode. Intuitively, with this approach the

classifier of an individual camera highly correlates with the camera view, including the types of activities in the field of view, and the view angle of the activities. So we expect this method to achieve the highest recognition rate among the three fusion methods since the observations are simpler in this case. However, labeling and training are required for each camera. In case that either camera deployment or room layout changes, the models need to be trained again.

*2) **Combined-view fusion**:* This is a feature-fusion approach similar to [8], [15]. For each episode, the bags of features from all cameras are concatenated sequentially to make a single feature vector for classification. For example, the feature vectors from cameras are $q_1, q_2, \ldots, q_M$, each of them is N-vector (codebook size is N). Then the concatenated feature vector is $Q = (q_1, q_2, \ldots, q_M)$, with length of MN (Fig. 3(b)). Vector Q is used for the SVM classifier. This fusion method is straightforward and it has the advantage of having more information (all the features from all cameras) for classification. The feature vector also encodes spatial information of the person, from the presence/absence of features in each camera's viewpoint. The major drawback is that feature Q depends on the cameras' deployment, which is a strong assumption on the system setup. But in practice, with a fixed smart home environment which does not change frequently (such as AIR lab), the system can be trained once and used for a long time. Intuitively, this approach makes use of all information available and should have a comparable performance to the best-view approach. However, it is possible that different views of different activities appear similar to each other thus causing more confusion between activities.

*3) **Mixed-view fusion**:* In this fusion method, a single activity classifier is obtained on the BoF of episodes (t seconds) irrespective of camera views (Fig. 3(c)). For each episode, when the number of spatio-temporal features is above a threshold meaning there is significant motion, the BoF of this episode will be used for activity classification. This approach gives a general activity model which does not depend on the camera field of view and the camera network setup. Therefore it is transferrable to the other environments. However, misclassification rate will be higher since now a single activity can be observed from significantly different view angles. This approach is quite different from the view-invariance approaches discussed in Sec. II, which all try to capture differences between viewpoints explicitly by modeling viewpoint as a latent parameter. In our mixed-view fusion, since we look at local motions and prune off observations with too few motions, we essentially eliminated views from the back of the person where the motions are occluded. So the effective views are mostly from profile to frontal ($\sim 180°$). It would be interesting to see whether the codebook and SVM are able to correctly represent such variations due to viewpoint changes.

In Sec. IV, we report experimental results of the above three fusion methods, and comment on the performances.

	cam 1	cam 2	cam 3	cam 4	cam 5
cutting	0	9032	0	0	0
scrambling	0	9042	0	0	0
eating	0	11397	11414	0	0
reading	3082	0	5176	10484	0
computer	4640	0	0	0	0
vacuuming	0	9052	0	0	17942
others	5130	9022	18160	8909	17902

TABLE II
NUMBER OF FRAMES FOR EACH ACTIVITY FROM EACH CAMERA.

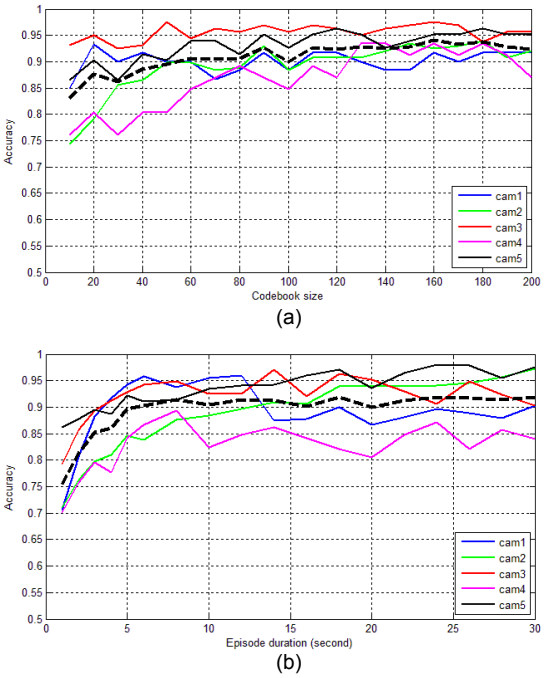

Fig. 4. Performance curve of best-view fusion. (a) Accuracy w.r.t. the codebook size. (b) Accuracy w.r.t. the episode duration. The black dashed line represents average accuracy of all cameras.

IV. EXPERIMENT

We acquired approximately 80 minutes of video sequences as the training data for the fine (second)-level of activity classification with spatio-temporal features (Fig. I). Each sequence is captured from all the five installed cameras, and two persons participated in the sequence taking. Performance of the coarse (first)-level activity classification can be found in [19]. Below we only present performance of the second-level activity analysis. The number of frames for each activity in each camera is shown in Table II. In all of the three fusion methods, we experimented on the codebook size(N) and the episode duration for collecting bag-of-features (t), in order to evaluate how they affect the classification accuracy. A three-fold cross-validation process is used in which when one fold is chosen as test data the other two are used for training. The three folds are randomly generated.

A. Performance of the three fusion methods

In best-view fusion, we report the performance for each camera without consideration of the best-view camera for

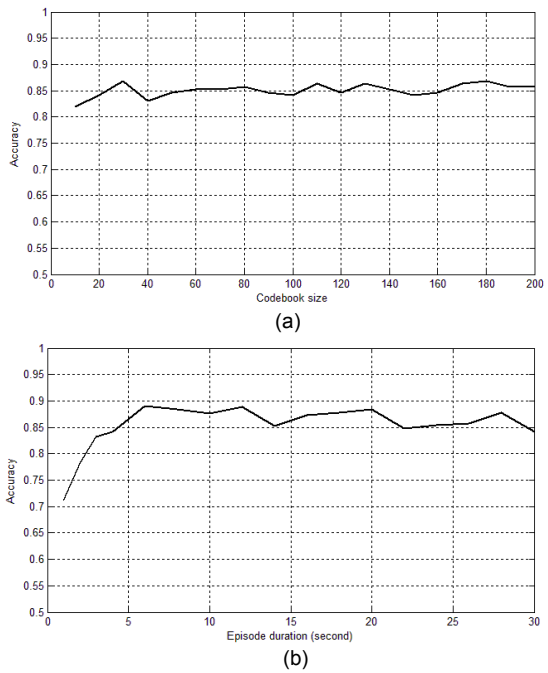

Fig. 5. Performance curve of combined-view fusion. (a) Accuracy w.r.t. the codebook size. (b) Accuracy w.r.t. the episode duration.

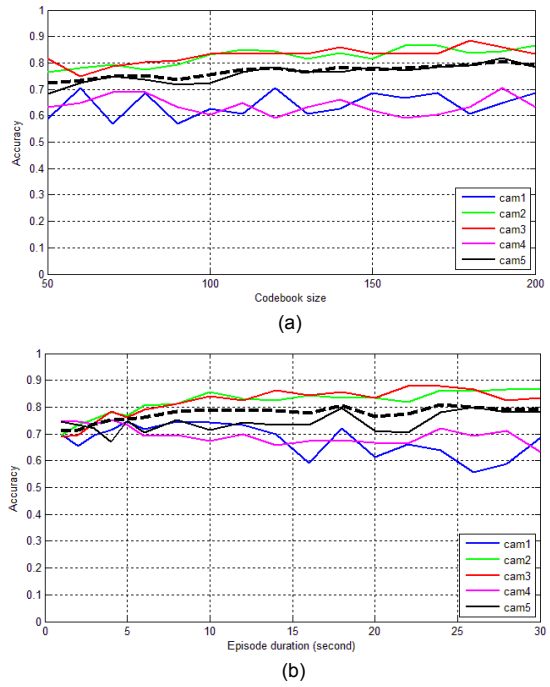

Fig. 6. Performance curve of mixed-view fusion. (a) Accuracy w.r.t. the codebook size. (b) Accuracy w.r.t. the episode duration. The black dashed line represents average accuracy of all cameras.

(a) Precision

	cam 1	cam 2	cam 3	cam 4	cam 5
cutting	–	0.92	–	–	–
scrambling	–	0.83	–	–	–
eating	–	0.96	0.98	–	–
reading	0.75	–	0.85	0.93	–
typing	0.95	–	–	–	–
vacuuming	–	0.95	–	–	0.91
others	0.94	0.78	0.98	0.76	0.95

(b) Recall

	cam 1	cam 2	cam 3	cam 4	cam 5
cutting	–	0.85	–	–	–
scrambling	–	0.85	–	–	–
eating	–	0.9	1	–	–
reading	1	–	0.92	0.76	–
typing	0.95	–	–	–	–
vacuuming	–	0.88	–	–	0.95
others	0.74	0.93	0.94	0.93	0.9

TABLE III
PRECISION AND RECALL OF CLASSIFICATION IN EACH CAMERA OF BEST-VIEW FUSION.

each episode, since there can be many ways for best-view selection which is not the major concern of this experiment. The best performance of all the cameras can be seen as an upper bound of the final performance after best-view selection, with the optimal condition being that the best-performance camera is always chosen as the best-view camera. Fig. 4(a) shows the accuracy of all activities for each camera, with codebook size ranging from 10 to 200, and episode duration being 15 seconds. Fig. 4(b) shows the accuracy of all activities for each camera, with episode duration from 1 second to 30 seconds, and codebook size being 100. Considering the average accuracy of all cameras, we observe when $N > 60$ and $t > 10$ seconds, the performance stays roughly stable. Classification accuracy for each activity in each camera can be found in Table III, when $N = 100$ and $t = 15$ seconds. The average accuracy of all cameras is slightly over 0.9.

For combined-view fusion, Fig. 5 shows the classification accuracy of all activities for different codebook size N (in (a), when episode duration is 30 seconds) and for different episode durations t (in (b), when codebook size is 100). Again, the accuracy is stable for wide range of N and t except when $N < 50$ or $t < 8$. In both Fig. 4(b) and Fig. 5(b), the curve has a big slope when episode duration is smaller than 5 seconds. This is because the episode can contain characteristics of the activity only when it is long than the repetitive cycle of the activity. The confusion matrix of classification can be found in Table IV, when codebook size is 100 and episode duration is 15 seconds. From Fig. 5 classification accuracy is around 0.85, and the curve is in general lower than that in Fig. 4.

Fig. 6 shows classification accuracy of mixed-view fusion. In Fig. 6(a), episode duration is fixed at 30 seconds. In Fig. 6(b), codebook size is fixed at 200. Table V is the confusion matrix when codebook size is 100 and episode duration is 15 seconds. Intuitively, accuracy would increase as codebook size increases, since for each activity there are

	typing	reading	eating	cutting	scrambling	vacuuming	others
typing	0.95	0	0	0	0	0.05	0
reading	0	0.61	0.03	0	0	0.03	0.33
eating	0	0	1	0	0	0	0
cutting	0	0	0	0.88	0.07	0	0.05
scrambling	0	0	0	0.02	0.88	0	0.1
vacuuming	0	0.02	0	0	0	0.88	0.1
others	0	0.05	0	0	0.02	0.04	0.89

TABLE IV
CONFUSION MATRIX OF COMBINED-VIEW FUSION, WHEN CODEBOOK SIZE IS 100 AND EPISODE DURATION IS 15 SECONDS.

	typing	reading	eating	cutting	scrambling	vacuuming	others
typing	0.5	0.07	0.07	0	0	0.14	0.22
reading	0.01	0.5	0.02	0.02	0.04	0.04	0.37
eating	0.01	0.01	0.86	0.03	0.01	0	0.08
cutting	0	0	0.01	0.75	0.11	0.01	0.12
scrambling	0	0.02	0.02	0.13	0.76	0.02	0.05
vacuuming	0.01	0.01	0.01	0.01	0	0.81	0.15
others	0	0.06	0.02	0.02	0.02	0.11	0.77

TABLE V
CONFUSION MATRIX OF MIXED-VIEW FUSION, WHEN CODEBOOK SIZE IS 100 AND EPISODE DURATION IS 15 SECONDS.

Fig. 7. Different views for *reading* from different cameras. It is challenging to recognize reading in the mixed-view approach since the viewpoints vary significantly.

different views. So the feature needs to be more descriptive. However in Fig. 6(a), the average accuracy of all cameras (black dotted line) is almost flat although slightly increasing. One reason can be because the cameras do not have wide overlapping field of views, so the same activity may not be observed by many cameras from many different view angles. In Fig. 6(b), we do not see the same slope as in Fig. 4(b) and Fig. 5(b) when episode duration is small. It is because to avoid confusion caused by small motion, episodes with small numbers of spatio-temporal features are omitted.

The results from mixed-view fusion show that classification accuracy is lower than 0.8, which is lower than best-view and combined-view fusion, and the activities are more easily confused with each other. This is expected since in this case an activity can be observed in different views and at different distance to the camera. Fig. 7 shows six different views of *reading* and the viewpoints vary significantly. Not only it is challenging for the codebook and classifier to learn the characteristic of the feature space with view variations, but also it has to differentiate from similar actions such as *scrambling*. Note that recognition rate of *typing* and *reading* is much lower than the previous two methods, and recognition rate of *reading* of combined-view is lower than that of best-view. On one hand, this is reasonable since *typing* and *reading* have more subtle motions and thus more challenging to recognize. On the other hand, however, notice that false classification of these two activities into *others* is high. From the pure purpose of activity recognition this is not satisfactory. But from the perspective of the application that builds on top of activity recognition in the smart home, misclassification into *others* does not cause any false input in high-level reasoning, which in the end may not affect the system's high-level functions as much.

B. Transfer of the activity classifier between camera views

The main advantage of the mixed-view approach over the best-view approach is that the activity classifier can be transferred to different camera views. This means that if we move the cameras or put them in another environment, the same activity recognition model can be used. We show some experiment results to demonstrate the performance gain. One issue here is that due to the limited overlapping field of view, two cameras usually have only one or two common activities in scene. Assume we want to test the classifier of camera i on camera j, and they have only one common activity a_k. Therefore, we want to see how $Classifier_i$ works on camera j in classifying a_k. After applying $Classifier_i$ on the samples from camera j, predictions are retained for a_k while other activities are considered $others$. This means we reduce activity classification on camera j into a two-class classification $\{a_k, others\}$, with the classifier from camera i. Accordingly, the confusion matrix from the mixed-view approach on camera j is converted into two classes. Precision and recall of a_k from the two approaches are compared in Table VI. The generic classifier from the mixed-view approach has a much better performance.

C. Summary of the fusion methods

The purpose of the experiments in this paper is to evaluate the three fusion methods for activity classification with multiple cameras, given their advantages and constraints on system setup and model transferability. There is no single criterion to chose from them. The tradeoff between classification accuracy, algorithm complexity and model transferability need to be considered. Table VII summarizes these tradeoffs.

	Transfer of the best-view model		Mixed-view performance	
	precision	recall	precision	recall
cam2 → cam3, eating	0.66	0.37	1	0.96
cam4 → cam1, reading	0.29	0.73	0.75	0.6
cam2 → cam5, vacuuming	0.52	0.78	0.75	0.86

TABLE VI
COMPARISON BETWEEN PERFORMANCE OF TRANSFERRING THE CLASSIFIER BETWEEN THE CAMERAS OBTAINED IN THE BEST-VIEW METHOD AND THAT OF THE MIXED-VIEW METHOD.

	Complexity	Transferability	Classification accuracy
Best-view	High. Need to a) train a model for each camera view; b) do camera selection.	Moderate. The activity model might be applied to cameras with similar views.	~ 0.9
Combined-view	Low. Since the ST-feature size is bigger, computation time is longer for training.	Low. It can only be applied to the same setup of multiple cameras.	~ 0.85
Mixed-view	Moderate. The activity classifier is the same for all cameras. But a final decision on activity needs to be made from activity classification results from individual cameras.	High. The activity classifier is trained irrespective of camera views and distance.	~ 0.8

TABLE VII
COMPARISON OF THE THREE FUSION METHODS.

V. CONCLUSION

In this paper we described a hierarchical approach for activity recognition in a home environment, given that there are many types of activities at home and it is more effective to recognize them with different image features and learning methods based on their characteristics. The second-level of activity recognition is described in detail, with the emphasis on the analysis of three camera fusion methods on the spatio-temporal feature based activity recognition. Experiments show that all the three methods yield reasonable performance on challenging activities. However, choosing one from them needs to consider the tradeoffs between the assumptions on system setup, model transferability and recognition rate. Comparative analysis of the tradeoffs is presented. Future work of multiview activity recognition in the home environment includes integrating all levels of activity recognition together and automatically segmenting the video stream for recognition.

REFERENCES

[1] E. M. Tapia, S. S. Intille, and K. Larson, "Activity recognition in the home setting using simple and ubiquitous sensors," in *Proceedings of PERVASIVE*, vol. 3001, 2004, pp. 158–175.

[2] M. Philipose, K. P. Fishkin, M. Perkowitz, D. J. Patterson, D. Fox, H. Kautz, and D. Hahnel, "Inferring activities from interactions with objects," *IEEE Pervasive Computing*, vol. 3, no. 4, pp. 50–57, 2004.

[3] B. Logan, J. Healey, M. Philipose, E. M. Tapia, and S. Intille, "A long-term evaluation of sensing modalities for activity recognition," in *Proc. of Ubicomp*, 2007.

[4] T. Moeslund, A. Hilton, and V. Kruger, "A survey of advances in vision-based human motion capture and analysis," *Computer Vision and Image Understanding*, vol. 103, no. 2-3, pp. 90–126, November 2006.

[5] J. Liu, J. Luo, and M. Shah, "Recognizing realistic actions from videos "in the wild"," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 2009.

[6] O. Brdiczka, J. L. Crowley, and P. Reignier, "Learning situation models in a smart home," *Trans. Sys. Man Cyber. Part B*, vol. 39, no. 1, pp. 56–63, 2009.

[7] J. Wu, A. Osuntogun, T. Choudhury, M. Philipose, and J. Rehg, "A scalable approach to activity recognition based on object use," in *IEEE Int. Conf. on Computer Vision*, 2007, pp. 1–8.

[8] S. Park and H. Kautz, "Privacy-preserving recognition of activities in daily living from multi-view silhouettes and rfid-based training," in *AAAI Symposium on AI in Eldercare: New Solutions to Old Problems*, 2008.

[9] F. Lv and R. Nevatia, "Single view human action recognition using key pose matching and viterbi path searching," in *CVPR*, 2007.

[10] D. Weinland, E. Boyer, and R. Ronfard, "Action recognition from arbitrary views using 3d exemplars," in *Proceedings of the International Conference on Computer Vision*, 2007, pp. 1–7.

[11] P. Natarajan and R. Nevatia, "View and scale invariant action recognition using multiview shape-flow models," in *CVPR*, 2008.

[12] P. Yan, S. M. Khan, and M. Shah, "Learning 4d action feature models for arbitrary view action recognition," in *CVPR*, 2008.

[13] R. Souvenir and J. Babbs, "Learning the viewpoint manifold for action recognition," in *CVPR*, 2008, pp. 1–7.

[14] A. Farhadi and M. K. Tabrizi, "Learning to recognize activities from the wrong view point," in *Proceedings of the 10th European Conference on Computer Vision*, 2008, pp. 154–166.

[15] G. Srivastava, H. Iwaki, A. Kosaka, J. Park, and A. Kak, "Distributed and lightweight multi-camera human activity classification," in *Third ACM/IEEE Conference on Distributed Smart Camera*, 2009.

[16] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local svm approach," in *ICPR*, 2004.

[17] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *ICCV VS-PETS*, 2005.

[18] J. C. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised learning of human action categories using spatial-temporal words," *International Journal of Computer Vision*, 2008.

[19] C. Wu and H. Aghajan, "User-centric environment discovery in smart home with camera networks," in *to appear*.

[20] I. Laptev, "On space-time interest points," *Int. J. Comput. Vision*, vol. 64, no. 2-3, pp. 107–123, 2005.