Real-time multi-view human action recognition using a wireless camera network

S. Ramagiri, R. Kavi, V. Kulathumani
Dept. of Computer Science and Electrical Engineering
West Virginia University
Email: {sramagir, rkavi}@mix.wvu.edu, vinod.kulathumani@mail.wvu.edu

***Abstract*—In this paper, we describe how information obtained from multiple views using a network of cameras can be effectively combined to yield a reliable and fast human action recognition system. We describe a score-based fusion technique for combining information from multiple cameras that can handle arbitrary orientation of the subject with respect to the cameras. Our fusion technique does not rely on a symmetric deployment of the cameras and does not require that camera network deployment configuration be preserved between training and testing phases. To classify human actions, we use motion information characterized by the spatio-temporal shape of a human silhouette over time. By relying on feature vectors that are relatively easy to compute, our technique lends itself to an efficient distributed implementation while maintaining a high frame capture rate. We evaluate the performance of our system under different camera densities and view availabilities. Finally, we demonstrate the performance of our system in an online setting where the camera network is used to identify human actions as they are being performed.**

I. Introduction

Real time recognition of human activities is increasingly becoming important in the context of camera based surveillance applications to quickly detect suspicious behavior and in the context of several interactive gaming applications. In this paper we design and evaluate a wireless camera network based action recognition system that can be used to classify human actions in real-time. While camera networks have the potential to increase the accuracy of action recognition by providing multiple views of the scene, using a multi-camera network for real time action recognition poses several challenges. We now describe these challenges and our contributions towards addressing them.

A. Contributions

1. Combining multi-view information: When using information from multiple views for action recognition we note that the angle made by the subject with respect to a camera while performing an action is not known. Pose estimation of a human subject based on body posture itself is a hard problem and it is therefore not feasible to assume that information. In this paper, we describe a fusion technique that effectively combines inputs from cameras capturing different views of a subject without knowledge of subject orientation. Moreover, the cameras acquiring data may not be deployed in any symmetry. It is not feasible to assume that entire 360 degree view for the action being performed is available. It is also infeasible to assume that camera configuration stays the same between the training phase and the actual testing phase. Only data from some viewing directions may be available and additionally some of these views may be partially occluded. We show how our fusion technique can seamlessly handle these cases. Using synchronous data collected from a network with up to 8 cameras, we analyze achievable system performance under different camera densities and view availabilities.

2. Local processing: In order to avoid overloading the network with too much data, it is important to ensure that individual frames are locally processed and only relevant data is sent to a fusion center for final classification [15]. At the same time, in the context of real-time recognition, it is equally important to keep the computational overhead low so that data can be locally processed at a high enough frame rate. Lower frame rates of processing will lead to lower data sampling rate and key motion information will be lost resulting in lower classification accuracy. Therefore, there is a need to avoid computationally expensive approaches for local feature extraction. In this paper we show how aggregated locality-specific motion information obtained from the spatio-temporal shape of a human silhouette, when used concurrently with information from multiple views using our fusion strategy, can yield good classification rates. Our feature descriptors use only size-normalized binary background subtracted human silhouette images, divide them into blocks and capture the spatio-temporal shape of individual blocks using first and second order image moments. Relying on such computationally simple operations for local processing lends itself to an efficient distributed implementation.

3. Real-time operation: We evaluate the performance of our system in a real-time setting where data extracted from the cameras is transmitted and actions are recognized while a subject is performing the different actions. In such a scenario, the start and end times for each action are unknown and also the data collected by the different cameras may not be perfectly synchronized in time. We implement our local feature extraction technique on an embedded camera network assembled using Logitech 9000 cameras and an Intel Atom processor based computing platform and use this system to recognize actions in a real-time setting.

B. Related work

Human action recognition has received significant research attention over the past several years [8], [13], [2] and many of these systems have exploited spatio-temporal features for action recognition [6], [10], [4], [7]. For example in [4] space-time shapelets based on human silhouettes are used for action recognition and in [7] the shape of a human silhouette over time is characterized for human action recognition. Our focus in this paper is on combining spatio-temporal shapes estimated from multiple views and performing this classification in real-time. In [12], spatio-temporal interest points are computed for an action being performed and a histogram based approach is applied that uses the number of spatio-temporal cuboids of each type in a region surrounding the interest points to classify

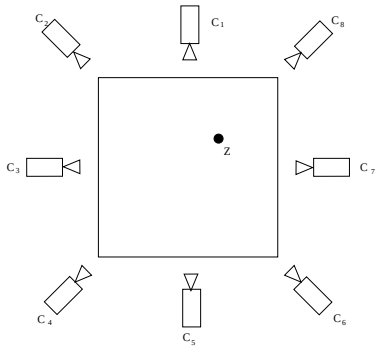

Fig. 1. The 8 camera layout for our experiments. The subject is shown at location z and can be anywhere within the square region. The relative camera orientations are assumed to be known but may not conform to the symmetric arrangement shown. Different subsets of cameras may be partially or fully occluded.

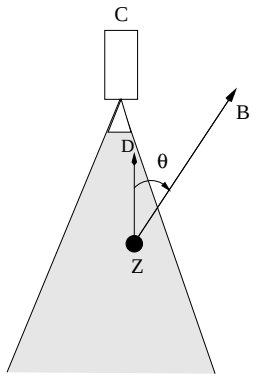

Fig. 2. View-angle θ of camera C with respect to action being performed. The subject is at point Z and performing an action while facing direction shown by ray ZB. The view-angle is measured in a clockwise direction from the ray ZD parallel to the optical axis of the camera.

individual actions. However, the histogram based approach loses information about the relative location of motion for each action, which is an important basis for classification in our approach.

Several multi-camera based approaches have also been proposed for action recognition [3], [1], [9], [11], [15], [12]. Some of these techniques such as [9] and [11] have used feature descriptors that are invariant to the view point for action recognition. By way of contrast, in this paper we train view-specific classifiers for each action. In [14], [16], a sequence of 3D visual hulls generated from multiple views have been used for action recognition. In [1], a human body model is used to extract feature descriptors that describe motion of individual body parts. A key difference of our work lies in the practical aspects of implementing a real time action recognition system. We have relied on computationally simpler operations that can provide higher frame rate for processing and also decrease communication overhead.

Fusion strategies for multi-view action recognition have been presented in [5]. In contrast to the best view classifier presented in [5], our technique uses data from all available views for classification and we highlight the robustness of our approach by considering cases when the best view(s) are not available. The view combination method presented in [5] and [12] combines feature vectors from individual cameras before performing the classification and this imposes a requirement on the configuration of cameras to be identical between the test and training phases. In contrast, we combine inputs from different views at a score-level by exploiting the relative orientation between cameras and as a result, we do not require the camera configuration to be preserved between training and test phases.

C. Outline of the paper

In Section 2, we describe our system model and problem statement. In Section 3, we describe our action recognition system. In Section 4, we evaluate the performance of our system. In section 5, we present conclusions and state future work.

II. Model and Problem Statement

Our system consists of N_C cameras that provide completely overlapping coverage of a region R from different viewing directions. The relative orientations between the cameras are assumed to be known, but the cameras may not conform to the symmetric arrangement shown in Fig. 1 and there may be fewer cameras. A subject can be performing any one of the following 10 actions in any (unknown) orientation with respect to the cameras: standing still, clapping hands, waving one arm (left or right), waving two arms, punching, jogging in place, jumping in place, kicking, bending and underarm bowling. We use $A_x, (1 \leq x \leq 10)$ to denote these actions. We have assumed that each action is performed at approximately the same pace and that there is only one subject within the region R at any given time. The subject can be at an any location within region R but this location is fixed for the duration of the action. The objective of the system is to recognize the action being performed based on data captured by the cameras at a sampling rate of f fps (frames per second).

In our specifc experimental setting, we use a network of up to 8 cameras deployed over a square region of 50 feet by 50 feet. The cameras are denoted as C_i $(1 \leq i \leq 8)$ and are deployed along the square region at a height of 8 feet from the ground. The subject can be at any location within this region such that each camera is able to capture the complete image of the subject (Fig. 1). We use the Logitech 9000 USB cameras for our experiments with data sampled at 20 fps and each frame captured at 640×480 resolution.

We define *view-angle* of a camera with respect to an action being performed as the angle made by the optical axis of the camera with the direction along which the subject performs the action sequence. View-angle is measured in the clockwise direction from the ray originating at the subject location that is parallel to the optical axis of the camera (Illustrated in Fig. 2). We divide the view-angle range of $0 - 360^o$ into N_v different sets by considering that different instances of the same action captured with small view-angle separations are likely to appear similar. For our specific experimental setting, we consider $N_V = 8$, but we note that N_v can be chosen independent of the number of the cameras in the system. The 8 view-angle sets are denoted as $V_j, (1 \leq j \leq 8)$ and are illustrated in Fig. 3 for camera C. For example, in Fig. 3, when the subject is facing the region between rays ZA and ZB, the camera C provides view V_2.

From here on, we say that a camera C_i provides view V_j of an action being performed if the view-angle of C_i with respect to the action being performed belongs to set V_j. At any given instant, it is not necessary that data from all views $V_j (1 \leq j \leq 8)$ are available. For instance, some cameras may not be active. It is also possible in certain deployments that the

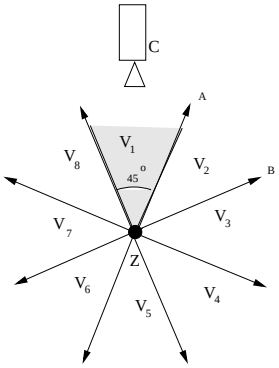

Fig. 3. The 8 view-angle sets for camera C with respect to the action being performed. The subject is located at point Z and could be facing any direction. The view-angles of camera C with respect to the action are grouped into 8 sets as shown in the figure. For example, when the subject is facing the region between rays ZA and ZB, the camera C provides view V_2.

angle between the principal rays of adjacent cameras are small enough and therefore the cameras provide the same views of the action being performed.

III. System Description

In this section, we describe our action recognition system. We divide our presentation into 3 parts: (1) extraction of feature descriptors, (2) collection of training data and (3) fusion of inputs from distributed cameras for action classification.

A. Extraction of local feature descriptors

Consider a window of F consecutive frames acquired by a camera that contain an action being performed. By subtracting the background from each frame, the silhouettes of the subject performing the action are extracted. A bounding box that envelopes all the background subtracted silhouettes is determined and drawn around each of the F extracted silhouettes as illustrated in Fig. 4. Only binary information is retained for each box (each pixel in the box is either part of the human silhouette or outside of it). The bounding boxes and the images within them are then normalized to a standard size of 300 pixels by 100 pixels. Each box is then subdivided into m smaller blocks to enable the charecterization of locality specific motion information. (In this paper we have divided each box into a 6 by 6 grid yielding 36 blocks). For each block in each of the F frames, we compute the zeroth, first and second order image moments that capture the shape of the silhouette. Using this feature descriptor we are able to characterize the locality specific motion information in each block of the human silhouette.

B. Collection of training data

In order to collect training data, videos of subjects performing each action are collected with the subject standing at the center of the square region and facing a reference camera (camera C_1 in Fig. 5). The actions are performed at different view-angles all of which belong to set V_1 with respect to the reference camera C_1. Because of symmetry, data collected in camera C_i corresponds to view V_i ($\forall i : 1 \leq i \leq 8$). 50 training action sequences are obtained for each action from all 8 views. Note that the symmetry is introduced only for ease of training data collection. During testing, the symmetry does not have to be maintained and the subject does not have to be at the center.

Once the training data is collected, we extract several 3

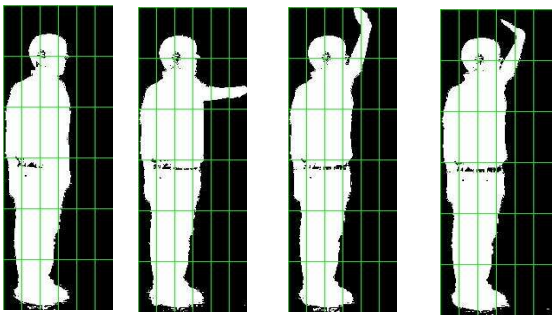

Fig. 4. Extracting local feature descriptors. A bounding box that encloses all background subtracted silhouette is drawn around each silhouette. The box is then scaled to a standard size of 300 by 100 pixels and divided into 36 blocks. Only binary information is retained for each block. For each block in each frame, we obtain the zeroth, first and second order image moments. Also, for each block we compute the average motion energy.

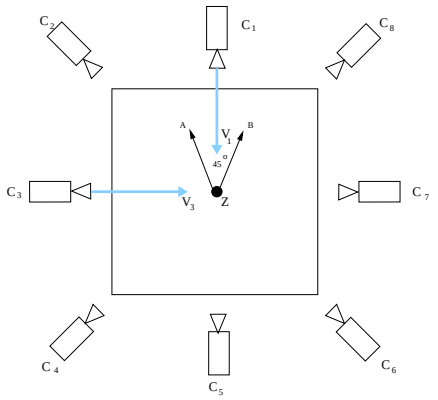

Fig. 5. Collection of training data. The subject is at the center of the network and performs each training action while facing the region between rays ZA and ZB. All cameras record the data. Because of symmetry, data collected in camera C_i corresponds to view V_i ($\forall i : 1 \leq i \leq 8$)

second windows from the training samples corresponding to each action and obtain feature descriptors as described in the previous subsection for each window. Corresponding to each action class and for each view, a two-class Linear Discriminant Analysis based projection vector is obtained by grouping together data belonging to that particular action against data from all other actions corresponding to the respective view. Let $\lambda_{a,j}$ correspond to the LDA projection vector corresponding to action $A_a (1 \leq a \leq 10)$ using data from view $V_j (1 \leq j \leq 8)$.

C. Action classification

We now describe how feature vectors are generated at each camera for a test action and how these are combined to generate a classification output. Consider that the subject performing a test action is at a point Z as shown in Fig. 6. Let the view provided by camera C_{ref} with respect to the action being performed be V_j (In Fig. 6, $ref = 1$). Note that V_j cannot be determined by C_{ref}. However, the angles $\theta_{r,s}$ between the principal axes of each pair of cameras (r, s) is known. And using $\theta_{ref,s} \quad : \quad (1 \leq s \leq N_c)$, relative to each of the N_v possible views $V_j (1 \leq j \leq N_v)$ that camera C_{ref} can provide for the action being performed, the views provided by all other cameras can be computed. This gives rise to a set ϕ of N_v possible configurations, which we denote as $\{\phi_k\}, 1 \leq k \leq N_v$. We let ϕ_k^i denote the view provided by camera C_i in configuration k.

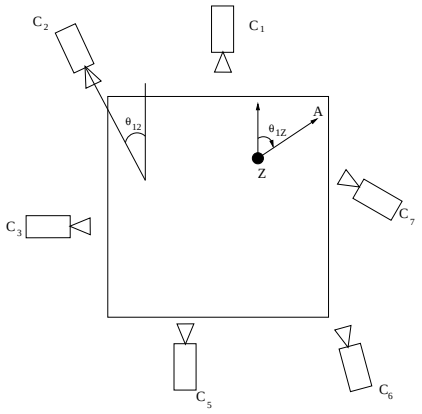

Fig. 6. Determining configuration set for subject location Z. Consider C_1 as the reference camera. The real orientation of the subject with respect to the reference camera is not known. But the angles $\theta_{r,s}$ between the principal axes of each pair of cameras (r, s) is known. Then for each possible view $V_j (1 \leq j \leq 8)$ that camera C_1 can provide for the action being performed, the views provided by other available cameras can be determined resulting in N_v possible configurations.

$$\phi = \{\{\phi_1\}, .., \{\phi_{Nv}\}\} \quad (1)$$
$$= \{\{\phi_1^1, .., \phi_1^{Nc}\}, .., \{\phi_{Nv}^1, .., \phi_{Nv}^{Nc}\}\} \quad (2)$$

Note that if the cameras retained the symmetric deployment during test, the 8 configurations would have followed a cyclical shift resulting in:

$$\phi = \{\{V_1, V_2, .., V_8\}, \{V_2, V_3, ..., V_1\}, .., \{V_8, V_1, .., V_7\}\}$$

However, the relative orientations between cameras need not be symmetric and two cameras r and s can provide the same view with respect to an action being performed if $\theta_{r,s}$ becomes very small. For illustration, suppose cameras C1 and C2 provide the same views with respect to a subject performing the action. In this case, the 8 configurations would be:

$$\phi = \{\{V_1, V_1, V_3, .., V_8\}, \{V_2, V_2, V_4, ..., V_1\}, ..\}$$

Note that in the scenario where certain cameras are absent or if their views are completely occluded, N_c now reflects the number of cameras from which data is available and each set ϕ_k in ϕ contains fewer number of elements.

Once the configuration set ϕ is determined, we use the feature descriptor generated from the test data at every camera to obtain matching scores under every configuration. This is done as follows. Consider score generation $S_{a,k,i}$ with respect to action A_a for data acquired by camera C_i under configuration ϕ_k. Let $\phi_k^i = j$, i.e., camera C_i is associated with view V_j under configuration ϕ_k. Let FV_i denote the feature vector computed for test data generated by camera C_i. In order to generate score $S_{a,k,i}$, we determine $FV_i \times \lambda_{a,j}$, calculate the distance of this product from the mean LDA score for action A_a, and then normalize the result to a range of $[0, 1]$ (0 indicates no match while 1 indicates perfect match). For each action A_s, $S_{a,k,i}$ represents the likelihood that test data from camera i corresponds to action A_a in vonfiguration ϕ_k. Similarly, a matching score is generated for each camera C_i in all configurations $\{\phi_k\}, 1 \leq k \leq 8$. If certain cameras have failed or occluded as shown in Fig. 6, then the matching scores corresponding to only the available cameras are computed under each configuration. For each action A_a, the net matching score $S_{a,k}$, which represents the likelihood that the test data from all cameras in the system corresponds to action A_a in configuration ϕ_k is computed as follows:

$$S_{a,k} = \sum_{i=1}^{Nc} S_{a,k,i} \quad (3)$$

After the configuration specific scores are generated, we compute the likelihood that the test data corresponds to action A_a by determining the maximum of $S_{a,k}$ over all configurations in the set ϕ. We denote this as S_a.

$$S_a = max(S_{a,k})_{k=1..8} \quad (4)$$

The action $A_F (1 \leq F \leq 10)$, with the highest score is classified as the action corresponding to the test data where F is determined as follows.

$$F = argmax(S_a)_{a=1..10} \quad (5)$$

IV. Performance evaluation

In order to systematically evaluate the performance of our action recognition system, we first present the classification performance using data collected by the 8 camera network and analyzed offline. Using this data, we evaluate the performance when views from different subsets of cameras are occluded including the case when data from cameras with most favorable views for identifying the specific action being performed are suppressed. Then we show how our fusion scheme can be altered to handle partial occlusions in all cameras, i.e., no camera has complete view of the subject but it is required that partial views available from a subset of cameras be combined for action recognition. Finally, we describe the implementation of our system on an embedded camera network and evaluate its performance n real-time where actions are classified as they are being performed.

A. Fewer cameras

We first collect test data from all the 8 cameras with different subjects performing a total of 20 test actions of each class, at different locations in the network. Then in order to evaluate the system performance with occlusions, we selectively use data from subset of cameras. We note down the ground truth for each test action to determine the classification accuracy for our system. In the offline analysis, we assume that each test action snippet is exactly 3 seconds that is entirely composed of the same action.

In Fig. 7, we plot the average classification accuracy of our system when different number of views are completely occluded. For the results shown in Fig. 7, the cameras whose data is unavailable are determined randomly. We note that classification accuracies are relatively high ($> 90\%$)with up to 4 cameras in the system.

We then analyze the robustness of our system by computing the clasification accuracies when data from cameras that are likely to yield the most suitable views for each action are not used. We order the favorable views for identifying each action based on the average motion energy observed in the training dataset for those actions from a given view. We define the average motion energy ($E_{a,j}^s$) for a given training sample s corresponding to an action A_a from a camera providing view v_j as the average number of pixels that have changed sign in sucessive frames of the training sample. Let $E_{a,j}$ denote the

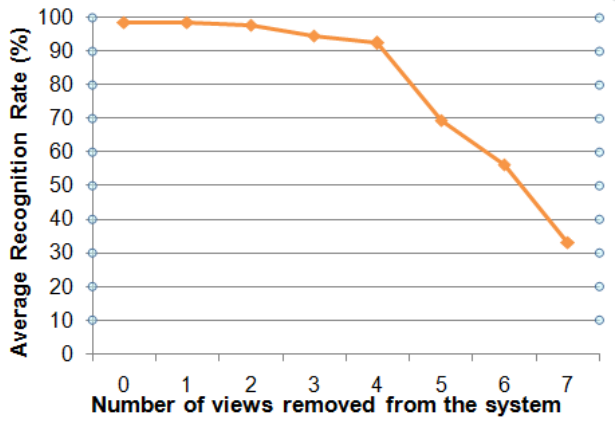

Fig. 7. Average classification accuracy of our system computed over test data from all actions when different number of views are completely occluded. The cameras whose data is unavailable are determined randomly.

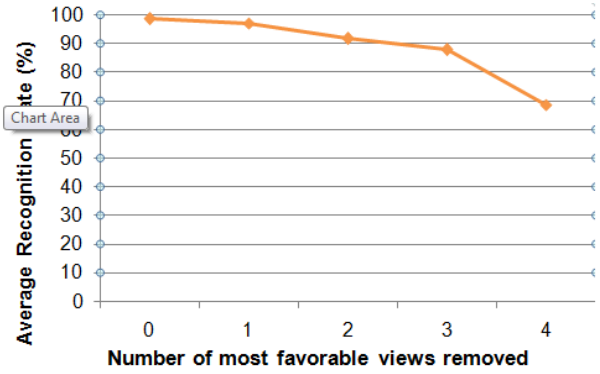

Fig. 8. Average classification accuracy of our system computed over test data from all actions when data from cameras providing 1 to 4 most favorable views for each action are suppressed.

average of $E^s_{a,j}$ computed over all training samples available for the corresponding action A_a and view V_j. Let ψ^a_i denote the ith favorable view for action a determined by ordering $E_{a,j}$ over all views, with ψ^a_1 being the most favorable. For each test action sample belonging to action A_a, we determine the respective cameras that provide these views. Fig. 8 shows the recognition accuracy when data from cameras providing 1 to 4 most favorable views for each action are suppressed. These results show that even when the best views are not available, information from other views is able to provide reasonably high recognition accuracy, thereby highlighting the significance of multi-view fusion. Accuracy of the system is observed to progressively decrease as more number of favorable views are removed.

B. Partial occlusions

In this subsection, we consider the case where some cameras have partial view of the subject but none have full view of the subject performing an action. To handle partial occlusions, we make an assumption that the regions of occlusion in each camera are known. Then we modify our fusion scheme in the following way to perform classification.

- Let $E_{a,j,b}$ denote the average motion energy per block b, based on all training data available for the corresponding action A_a and view V_j.
- Let $\{B_{a,j}(s)\}$ represent the set of top $s\%$ of the blocks in terms of the motion energy for action A_a under view V_j. In the presence of partial occlusions at camera C_i, the data from C_i is used for determining the match score for action A_a under view V_j only if none of the blocks in $\{B_{a,j}(s)\}$ are occluded.
- Suppose data from C_i is used for determining the match score for action A_a under view V_j, then we retrain our classifier to obtain $LDA'_{a,j}$ based only on training data from blocks in the set $\{B_{a,j}\}$. We use $LDA'_{a,j}$ to determine the match scores. For the results presented in Fig. 9, we set $s = 50$.

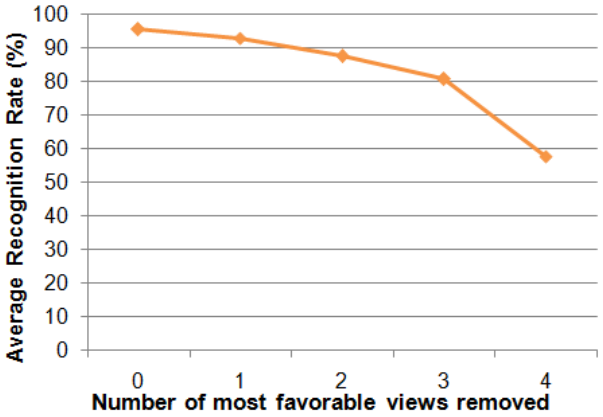

Fig. 9. Partial occlusions: Average classification accuracy of our system computed over test data from all actions when data from cameras providing 1 to 4 most favorable views for each action are suppressed and only 50% of the data is available from the other cameras.

We note that if the regions of the silhouette that are most pertinent for recognizing a given action are blocked from all available views, then it is difficult to recognize that action. The objective of this particular experiment is to evaluate the achievable recognition rate, when data suitable for recognizing a given action is available at least from a partial set of cameras. For the results shown in Fig. 9, we retain only the data from the blocks in $\{B_{a,j}(50)\}$ and occlude all other data for each action a and view V_j. We then additionally suppress all data from 0 to 4 most favorable views and determine the recognition accuracies shown in Fig. 9.

C. Real-time performance evaluation

In order to evaluate the real-time performance of our system, we implemented the background subtraction and local feature descriptor extraction on an embedded camera network assembled using Logitech 9000 cameras and an Intel Atom processor based computing platform equipped with an 802.11 wireless card for communication. We deployed a system of $N_c = 5$ cameras over the 50 by 50 region (specifically camerass C1, C3, C5, C6, C7 in Fig. 1). Each subject performs one of the actions stated in Section II for a duration of 10 seconds, then switches to another action from the list for 10 seconds and this is repeated 5 times for each subject.

The distributed cameras perform local processing on an analysis window of 3 seconds. However, it is important to note that the start of an action may not exactly align with the start of an analysis window. Since an action can happen at any time and the system needs to be continuously acquiring and processing data, we apply the feature vector extraction on analysis windows with an overlap of 2 seconds. In other words,

we only move the window by 1 second each time. The embedded cameras run the NTP protocol for clock synchronization in order to enable the computation of feature descriptors by different cameras over synchronous time windows. However, we empirically note a time synchronization error of approximately 50 mS between the different cameras.

	Recognition accuracy (%)
Steady state	85.2
Transition state	70.1

Fig. 10. The classification accuracy for the real-time system separately for the steady and transition states in each action sequence: The 2 seconds interval at the start of each new action sequence as the transition state.

Once the feature vectors are computed at camera i for every 1 second, the scores $S_{a,k,i}$ are determined corresponding to each action A_a in every configuration ϕ_k that camera i can belong to. Only these scores are transmitted wirelessly to a fusion center. The LDA vectors $\lambda_{a,j}$ for each action A_a and view V_j are pre-computed and stored on each camera to enable the computation of scores $S_{a,k,i}$. Thus in our system with 10 actions and trained for 8 view-angles, we transmit only 320 bytes per second from each camera (where each score is represented as 4 bytes). The feature descriptors are then combined at the fusion center based on our score-based fusion technique. A classification result is generated every 1 second, but with a 2 seconds lag to ensure that all data corresponding to a given interval has arrived at the fusion center before processing data for that interval.

In Table 10, we have presented the classification accuracy for the real-time system separately for the steady and transition states in each action sequence. We have defined the 2 seconds interval at the start of each new action sequence as the transition state. We notice that during the transition phase when moving between actions we notice higher mis-classifications but in the steady state we observe significantly fewer mis-classifications.

V. Conclusions and Future Work

In this paper, we have described a score-based fusion strategy to combine information from a multi-view camera network for recognizing human actions. By systematically collecting training data from different views for an action and combing data from cameras at a score-level, we are able to accomodate arbitrary camera orientations during the testing phase. We applied our fusion technique on view specific classifiers designed over computationally simple feature descriptors obtained at each camera that capture the spatio-temporal shape of a human action. We note that the fusion technique described in this paper for combining data from multiple views can also be applied in conjunction with view-specific classifiers obtained using feature descriptors other than the ones used in this paper.

We tested the performance of our system using data collected from an 8 camera network. Our system is able to achieve an accuracy of 95% in classifying actions when all cameras are present. We showed that our system can tolerate non-availability of data from cameras that provide the *best* views for classifying a given action. We also described how the locality-specific feature descriptors enable our system to handle partial occlusions. We then tested the performance of our system in an online setting by implementing our feature extraction and fusion technique in an embedded camera network and by classifying actions as they are being performed.

In future, we would like to relax our assumptions with respect to knowledge of occluded areas of a silhouette and instead plan to use the information from multiple views to reliably detect occluded regions of a foreground object. We also plan to apply our action recognition system towards recognizing long duration activities and activities involving more than one subject by modeling the sequence of estimated actions by our system.

References

[1] H. Aghajan and C. Wu. Layered and collaborative gesture analysis in multi-camera networks. In *International Conference on Acoustics, Speech and Signal Processing*, 2007.

[2] M. A. R. Ahad, J. Tan, H. Kim, and S. Ishikawa. Human activity recognition: Various paradigms. In *International Conference on Control, Automation and Systems*, pages 1896 – 1901, 2008.

[3] O. Akman, A. A. Alatan, and T. Ailoglu. Multi-camera visual surveillance for motion detection, occlusion handling, tracking and event recognition. In *Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications*, 2008.

[4] D. Batra, T. Chen, and R. Sukthankar. Space-time shapelets for action recognition. In *IEEE Workshop on Motion and Video Computing*, pages 1–6, 2007.

[5] C. Wu and A. Khalili and H. Aghajan. Multiview activity recognition in smart homes with spatio-temporal features. In *International Conference on Distributed Smart Cameras (ICDSC)*, 2010.

[6] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 65–72, 2005.

[7] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *International Conference on Computer Vision*, pages 1395–1402, 2005.

[8] W. Hu, T. Tan, L. Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems,Man and Cybernetics*, 34:334–352, 2004.

[9] P. Natarajan and R. Nevatia. View and scale invariant action recognition using multiview shape-flow models. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2006.

[10] A. Oikonomopoulos, I. Patras, and M. Pantic. Spatiotemporal salient points for visual recognition of human actions. *Systems, Man and Cybernetics*, 36(3):710–719, 2005.

[11] V. Parameswaran and R. Chellappa. View invariance for human action recognition. *International Journal of Computer Vision*, 66(1):83–101, 2006.

[12] G. Srivastava, H. Iwaki, J. Park, and A. C. Kak. Distributed and lightweight multi-camera human activity classification. In *Third ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)*, pages 1–8, 2009.

[13] P. K. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. *IEEE Transations on Circuits, Systems and Video Technologies*, 18(11):1473–1488, 2008.

[14] D. Weinland, R. Ronfard, and E. Boyer. Free viewpoint action recognition using motion history volumes. *Computer Vision and Image Understanding*, 104(2-3):249–257, 2005.

[15] C. Wu, H. Aghajan, and R. Kleihorst. Real-time human posture reconstruction in wireless smart camera networks. In *International Conference on Information Processing in Sensor Networks (IPSN)*, pages 321–331, 2008.

[16] P. Yan, S. Khan, and M. Shah. Learning 4d action feature models for arbitrary view action recognition. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–7, 2008.