On efficient use of multi-view data for activity recognition

Tommi Määttä  
University of Technology  
Eindhoven, The Netherlands  
t.maatat@tue.nl

Aki Härmä  
Philips Research  
Eindhoven, The Netherlands

Hamid Aghajan  
Ambient Intelligence Research Lab  
Stanford University

ABSTRACT

The focus of the paper is on studying five different methods to combine multi-view data from an uncalibrated smart camera network for human activity recognition. The multi-view classification scenarios studied can be divided to two categories: view selection and view fusion methods. Selection uses a single view to classify, whereas fusion merges multi-view data either on the feature- or label-level. The five methods are compared in the task of classifying human activities in three fully annotated datasets: MAS, VIHASI and HOMELAB, and a combination dataset MAS+VIHASI. Classification is performed based on image features computed from silhouette images with a binary tree structured classifier using 1D CRF for temporal modeling. The results presented in the paper show that fusion methods outperform practical selection methods. Selection methods have their advantages, but they strongly depend on how good of a selection criteria is used, and how well this criteria adapts to different environments. Furthermore, fusion of features outperforms other scenarios within more controlled settings. But the more variability exists in camera placement and characteristics of persons, the more likely improved accuracy in multi-view activity recognition can be achieved by combining candidate labels.

1. INTRODUCTION

As people move around in an environment covered by multiple smart cameras, they can be seen in varying number of views depending on the geometry of the place and relative location of subjects and cameras. A single camera might provide enough information to classify the activity of a person with a reasonable accuracy, but having other view(s) providing simultaneously data from other perspectives is expected to improve the accuracy. The information about the activities and behaviour of the subjects can be used to control various services, e.g., in smart homes or shops, or to collect important events, e.g., in care homes or hospitals.

One proposed approach to handling multiple views has been to explicitly account for the change in aspect/view. A classifier trained in one view should be put to use in a completely new view. By looking into feature correspondences within the same classes of activities between the source view and the target view a suitable transfer of the model could be found. These methods proposed in [4] and [5] are computationally expensive and require a fixed person-to-camera angle to work optimally. [2] presents that view-invariant recognition can be performed by using a simple data fusion of minimum of two orthogonal views. This is based on the knowledge that better recognition is achieved e.g. from a side view for a kicking or pointing action. Therefore, by using the appropriate camera to recognize a particular activity should provide better accuracy. In real-world scenarios assumption of view orthogonality becomes easily not valid. And choosing appropriate view is driven by the type of action, which itself is unknown too.

The approach proposed in this paper aims to provide nearly view-independent image features. All the views are given the same priority and no geometrical assumptions are made. Therefore, our approach aims to develop a single model that works with all the views close to optimal, in practice unfeasible, view-specific models. There are two basic approaches in combining multi-view data for activity classification. Either each camera performs individually the classification and these labels are centrally combined into a single class decision, or each view provides feature information to be centrally processed. The centralized process can, e.g., select the best source of information, or combine the given features into a single feature representation.

In this paper we will start by providing the details on the methodologies used in this study for modeling human activities and for combining data from multiple views according to described multi-camera use scenarios. After describing the camera scenarios, we continue to explore how these different scenarios perform. We finish by discussion on the results.

2. ACTIVITY MODELING

For classifying activities based on video, the video has to be processed to a simpler form that focuses on the person and gives essential information. As the vision networks under study are uncalibrated, each view is independently processed. The output from a view is either a stream of features or labels depending on whether classification is done within the smart camera or by a centralized process.

2.1 Image features

Regardless of its challenges silhouette extraction is a widely
used method in pre-processing images to be able to focus on the object of interest, in our case people. Silhouettes themselves could already be used as representation of people. But due to various artifacts depending on selected foreground segmentation method, we applied background subtraction \( \mathcal{S} \), silhouettes are usually processed further into more robust and simpler descriptions, called image features.

We tested three fundamental categories which contain 29 silhouette-based features combined. First, assuming a relatively upright cameras, person’s posture would be a vital clue. Second, global motion of a moving person would offer insight into how mobile the person is. And third, local motion of the person would indicate an activity involving the use of person’s limbs \( \mathcal{S} \) or minor shifts of balance/orientation. The features are listed in Table 1 and in Figure 1, one can see illustrations on the four basis descriptions used.

In this paper all features representing changes over time are transferred from frame-to-frame differentials to the change of a characteristic in seconds. This is obtained by multiplying differentials by scaling factor \( s \). Factor \( s \) is defined as the ratio of capture frequency \( fps \) and length of the observation window \( \text{w} \):

\[
\begin{align*}
\text{w} & = \frac{\text{fps}}{\text{l}w} \quad (1)
\end{align*}
\]

In addition to compensating for temporal differences between environments, when having fixed window length, a mean-variance normalization to each image feature was performed. For each dataset the mean \( \mu_j \) and variance \( \sigma_j \) of each feature \( v_j \) is computed and used to center and scale the distribution of each feature. Goal is to normalize the features into a form that generalizes them better to other environments/datasets.

\[
\begin{align*}
v_{j}^{\text{new}} & = \frac{v_j - \mu_j}{\sigma_j} \quad (2)
\end{align*}
\]

Figure 1: Example on the silhouette-based computed features used as basis, starting from left) bounding box, principal angle, median coordinate and 4-bin shape descriptor.

### 2.1.1 Posture

To detect the posture and changes in posture two fundamental features were exploited; bounding box aspect ratio and principal component angle. The aspect ratio of a bounding box surrounding the silhouette pixels was used as a rough measure of posture. In addition to using the aspect ratio of the bounding box \( AR_{bb} \), also its behaviour over time was studied. We focused on observing the change in the aspect ratio \( \Delta AR_{bb} \), box height \( \Delta H_{bb} \), and box width \( \Delta W_{bb} \).

The overall posture was characterized by the angle of the silhouette on the image plane. This angle was estimated using principal component analysis and marked hereafter as \( \text{PCA} \). The behaviour of the angle introduced two additional features: the change-in-angle \( \Delta \text{PCA} \) and the change of change-in-angle \( \Delta^2 \text{PCA} \).

#### 2.1.2 Global Motion

To detect global motion the overall movement of the 2D silhouette was used. A median coordinate of the positions of all the silhouette pixels \( mc \) is used as the basis. By observing the first and second differential of the movement of median coordinate we were able to detect global motion \( \Delta \) and acceleration \( \Delta^2 \) in both horizontal \( W \) and vertical \( H \) directions. For additional global motion cue the bounding box location change in horizontal \( \Delta LW_{bb} \) and vertical directions \( \Delta LH_{bb} \) was also computed.

#### 2.1.3 Local Motion

To detect minor changes in the positions and orientations of limbs, a 4-bin representation of a vertical histogram on silhouette pixels was computed, called hereafter the Shape Descriptor \( SD \). In addition to representing how the mass of the person is distributed in vertical direction, the behaviour of \( SD \) was also studied. The deformation \( \Delta SD \) and deformation of deformation \( \Delta^2 SD \) were used to describe how the mass distribution evolves over time.

In addition to detecting the behaviour of the median coordinate on the image plane, the same observations were performed wrt the bounding box \( mc bb \). This results in similar coordinate differential features, but now observed within the bounding box, as cue for local motion.

### 2.2 Classifier structure

As the class categorization under consideration seemed to have a natural way to conform under a hierarchical model, tree was taken as the structure for the classifier. The tree would be binary, only two splits per node. The root split discriminates between mobile (walk/run) and stationary activities (others). The mobile branch is split again into classes representing walking and running activities. The stationary activities are split into upright (stand/stand-interact) and less upright categories, after which decision on interactivity of the action is made, see Figure 2.

Figure 2: Logical tree structure for the six activities of interest.

### 2.3 Temporal models
Table 1: Image Features in 3-categories: Posture and Global and Local Motion, based on 4 representations: Bounding Box, Principal Angle, Median Coordinate and Shape Descriptor.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Index</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bounding Box</td>
<td>1</td>
<td>(AR_{bb}(t) = \text{height}<em>{bb}(t) \times \frac{\text{width}</em>{bb}(t)}{2})</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>(\Delta AR_{bb}(t) = s \cdot (AR_{bb}(t) - AR_{bb}(t-l_w)))</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>(\Delta X_{bb}(t) = s \cdot \left[\frac{X_{bb}(t) - X_{bb}(t-l_w)}{X_{bb}(t)}\right]) with (X = H, W)</td>
</tr>
<tr>
<td>PC Angle</td>
<td>2</td>
<td>(PCA(t) = \begin{cases} \arctan(PCA(t)/PC_H(t))/Pi \times 180 &amp; \text{if } \arctan(PCA(t)/PC_H(t)) \geq 0 \ 180 - \arctan(PCA(t)/PC_H(t))/Pi \times 180 &amp; \text{if } \arctan(PCA(t)/PC_H(t)) &lt; 0 \end{cases})</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>(\Delta PCA(t) = s \cdot (PCA(t) - PCA(t-l_w)))</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>(\Delta^2 PCA(t) = s \cdot ((PCA(t) - PCA(t-l_w)) - (PCA(t-l_w) - PCA(t-l_{w}))))</td>
</tr>
<tr>
<td>Global Motion</td>
<td>3</td>
<td>(\Delta LX_{mc}(t) = s \cdot \left[\frac{X_{mc}(t)}{X_{mc}(t-l_w)}\right]) with (X = H, W)</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>(\Delta^2 LX_{mc}(t) = s \cdot \left[\frac{X_{mc}(t)}{X_{mc}(t-l_w)}\right]) with (X = H, W)</td>
</tr>
<tr>
<td>Median Coordinate</td>
<td>10</td>
<td>(\Delta LX_{mc}(t) = s \cdot \left[\frac{X_{mc}(t)}{X_{mc}(t-l_w)}\right]) with (X = H, W)</td>
</tr>
<tr>
<td>Local Motion</td>
<td>11</td>
<td>(\Delta LX_{mc}(t) = s \cdot \left[\frac{X_{mc}(t)}{X_{mc}(t-l_w)}\right]) with (X = H, W)</td>
</tr>
<tr>
<td>Shape Descriptor</td>
<td>14</td>
<td>SD_1, vertical silhouette scanline quantized to 4-bins centered on mc</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>SD_2</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>SD_3</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>SD_4</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>(\Delta SD_1(t) = s \cdot (SD_1(t) - SD_1(t-l_w)))</td>
</tr>
<tr>
<td></td>
<td>19</td>
<td>(\Delta SD_2(t))</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>(\Delta SD_3(t))</td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>(\Delta SD_4(t))</td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>(\Delta^2 SD_1(t) = s \cdot \left[(SD_1(t) - SD_1(t-l_w)) - (SD_1(t-l_w) - SD_1(t-l_{w}))\right])</td>
</tr>
<tr>
<td></td>
<td>23</td>
<td>(\Delta^2 SD_2(t))</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>(\Delta^2 SD_3(t))</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>(\Delta^2 SD_4(t))</td>
</tr>
<tr>
<td>Median Coordinate BB</td>
<td>26</td>
<td>(\Delta LX_{mcbb}(t) = s \cdot \left[\frac{X_{mcbb}(t)}{X_{mcbb}(t-l_w)}\right]) with (X = H, W)</td>
</tr>
<tr>
<td></td>
<td>27</td>
<td>(\Delta^2 LX_{mcbb}(t) = s \cdot \left[\frac{X_{mcbb}(t)}{X_{mcbb}(t-l_w)}\right]) with (X = H, W)</td>
</tr>
</tbody>
</table>

A 1D model, that could describe the patterns in the features, was needed in order to decide which branch of the classification tree to take in each of the nodes. From widely used temporal models such as hidden Markov models (HMMs) [10] and maximum entropy Markov models (MEMMs) [9], we used conditional random fields (CRF) [7] as representative of these models. See Figure 3 for graph on the 1D chain structure of HMM and CRF models.

HMM would assume that process being modeled is a Markov process, current label \(Y_t\) depending only on the previous hidden state \(Y_{t-1}\) and the current observation \(X_t\). Whereas, CRF conditions on the entire observation sequence to find the most likely label sequence. CRF does not model the properties of the observations themselves like HMM does and by not having independence assumptions on the observations it should be able to incorporate bigger set of features than HMM. After experimentation, the observation interval \(l_w\) was fixed to 5-frames for all the experiments reported in this paper. For model training, iterations were limited to 600 and convergence was guaranteed by re-training in case of badly fitting model.

3. Multi-View Scenarios

We introduce five different camera use scenarios to get insight into how multi-camera systems should handle multiple streams of image features of the same scene, see Figure 3 for illustration. We defined the different camera scenarios in a general manner so that further tailoring can be done if additional information of the environment is available.

The camera scenarios are divided into two distinct categories. Scenarios SingleView(SView), KeyView(KView) and OracleView(OView) are methods that rely on features given by available or selected single view. Whereas PresentViews (PViews) and CombiView(CView) scenarios are based on fusion of multi-view data either on label- or feature-level. A view that observes a person (POV) is used to refer to a view that has the person in its field of view (FOV). Person might be only partially visible in a POV. For each time instant \(t\)
the amount of POVs is given by \( n \), and the feature-vector of size \( fL \times 1 \) of view \( i \) is given by \( v_i(t) \).

### 3.1 View selection

SingleView represents a case when only one view is observing the person. SingleView uses thus one fixed view, no fusion of features or labels necessary. The feature-vector \( f_{SV}(t) \) with fixed view \( i \) as input for classification is given by:

\[
f_{SV}(t) = v_i(t)
\]

KView is defined as the POV that fills the silhouette-based three-part criteria the best. The three parts in descending priority: silhouette fully in FOV (not cut off), size of the area three-part criteria the best. The three parts in descending priority: silhouette fully in FOV (not cut off), size of the area

The feature-vector \( f_{KView}(t) \) is thus given by:

\[
f_{KView}(t) = F\{v_1(t)v_2(t)\ldots v_n(t)\}_{fL\times n} = F\{V(t)\}
\]

KView is defined as the POV that fills the silhouette-based three-part criteria the best. The three parts in descending priority: silhouette fully in FOV (not cut off), size of the area

\[
f_{KView}(t) = \sum_{i=1}^{n} I_{bestView}(v_i(t))v_i(t)
\]

OView is a theoretical scenario, in which the system would be able to pick the POVs that would give the correct label for activity, if such a view existed for a given time instant. This scenario gives us a reference on how an ideal single camera selection, ideal KView, would perform with the given classifier system.

### 3.2 View fusion

All the scenarios defined so far have been based on the use of a single view, either by choice or circumstance. PViews scenario uses all of the POVs individually to first classify and only after combines the labels. The classifier input \( f_{PView}(t) \) is now a set of all the POV feature-vectors:

\[
f_{PView}(t) = \{v_1(t), \ldots, v_n(t)\}
\]

And the output class label \( c(t) \) is the most common label \( L \) among all the individual classifier outputs \( c_i(t) \).

\[
c(t) = \arg \max_L \sum_{i=1}^{n} I_{equalToL}(c_i(t))
\]

As PViews combined label-data, CView scenario in the other hand combines features of all the POVs into a single feature-vector. Feature fusion is performed according to a multiplication with a transformation vector \( K \). In our approach \( K \) was set as an averaging transformation, but other operations such as weighted average could provide improvement. The input feature-vector \( f_{CView}(t) \) is therefore defined as:

\[
f_{CView}(t) = [v_1(t)v_2(t)\ldots v_n(t)]_{fL\times n} \cdot K
\]

\[
K = \frac{1}{n} \begin{bmatrix} 1 & 1 & \ldots & 1 \end{bmatrix}_1 \times fL
\]

### 4. EXPERIMENT DATA: MAS, VIHASI AND HOMELAB

In this paper three different multi-view datasets are used. Two of them, MAS (Manually Annotated Silhouettes) and VIHASI (Virtual Human Action Silhouettes) offer ideal silhouettes [1]. Silhouettes are without any holes, noise blobs or other artifacts with all the cameras observing the person at all times. MAS is based on recorded people performing actions live in their own pace. VIHASI is based on motion-capture driven avatars, whose appearance differs, but the actions they perform are exactly the same in duration and pace. Third dataset is called HOMELAB, as it was recorded in a multi-room laboratory space furnished as a home environment, see Figure 4 for examples of the environment and views. In HOMELAB dataset ten subjects (heights between 150 and 190cm) perform the same scripted routine in their own pace. See Table 2 for more specific information on the different datasets and Figure 5 for their actors.

As only the HOMELAB dataset had been recorded and annotated with the 5 main activities in mind, both MAS and VIHASI datasets had to be reannotated to match the already existing activity categories. Reannotation was done by merging some primitive activities under a existing activity label and in some cases the primitive activity had to partly or entirely removed from the new reannotated dataset. Table 3 portrays how the original activities of both datasets were handled to fit them under the existing activity classes. In addition to the 5 classes of HOMELAB data a 6th class for running activity was introduced.

### 5. EXPERIMENTS: FEATURE SELECTION

To observe the dependence of features on the environment where they are computed, we studied how different features
Table 3: 15 action classes from VIHASI and 12 from MAS are mapped to the 6 activities.

<table>
<thead>
<tr>
<th>New Class</th>
<th>Original VIHASI Classes</th>
<th>Original MAS Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Walk:</td>
<td>HeroOpenDoor, Walk, WalkTurn180(partly)</td>
<td>WalkToLeft, WalkToRight</td>
</tr>
<tr>
<td>Stand:</td>
<td>Collapse(partly), StandLookAround</td>
<td>CollapseLeft(partly), CollapseRight(partly), StandUpLeft(partly), StandUpRight(partly)</td>
</tr>
<tr>
<td>Sit:</td>
<td>HeroSmash, Stand, Punch</td>
<td>GuardToKick, GuardToPunch, KickRight, PunchRight</td>
</tr>
<tr>
<td>SitInt:</td>
<td>KnockOutSpin(partly)</td>
<td>CollapseLeft(partly), CollapseRight(partly), StandUpLeft(partly), StandUpRight(partly)</td>
</tr>
<tr>
<td>Run:</td>
<td>Run, RunPullObject, RunPushObject</td>
<td>RunLeft, RunRight</td>
</tr>
</tbody>
</table>

Figure 5: Example on actors, Top-row) all five virtual actors of VIHASI dataset, bottom row from left) Actor1 running and Actor4 StandingInteracting both from all 2 views of MAS dataset.

Figure 6: Examples on the 8 partly overlapping views of the HOMELAB dataset with a person taking a drink in kitchen (views 1,3,4,6), and later person putting the drink down on the sofa table (views 2,5,7,8).

correlated within each dataset. The correlation between all the 29 features within the three datasets is shown in Figure 4 as separate correlation matrices. Some significant correlation, above 0.8, exist in all of the datasets between few different image feature-pairs. Correlation is evident between features that are closely connected to each other in computation, such as change of a bounding box dimension and change of aspect ratio. No other strong correlations exist, but is does depend on the dataset which feature-pairs show this strong correlation.

5.1 Feature selection with PFA

Based on results shown in Figure 7 it is feasible to find feature subsets with low-correlation within a dataset. A technique called Principal Feature Analysis proposed in [8] was used to find the candidate subsets of features. PFA aims to select features that maintain as much as possible of the variability and spread of the original data given a number of features based on principal component analysis. The candidate feature subsets were computed for nine different numbers of features: 3, 5, 7, 9, 11, 13, 15, 17, and 29, and for each of the datasets and the combination dataset. The correlation between the PFA selected candidates of entire HOMELAB dataset is shown in Figure 8(a) with corresponding examples for the entire MAS and VIHASI datasets in Figure 8(b) and Figure 8(c).

The PFA-selected subsets offer reduced correlation. For all the entire datasets a subset with no significant correlation, > 0.6, could be observed until 9 features when increasing the number of features starting from 3. Between HOMELAB and MAS datasets some overlap in selected features exist, but when examining all three datasets, the overlap is smaller. A different subset of features should be used for each of the environments to provide optimal accuracy in classification.

5.2 Node-specific selection

The optimization of the number and type of features used in each of the splits in the classification tree can provide better accuracy in classification. The PFA can also be performed individually in each branch of the classification tree. The resulting candidates are then compared to each other by examining the fit of the model, trained by the candidate subset, to the training data. The candidate whose model fits the training data the best is expected to provide similar accuracy with new data. See Figure 9(a) for the performance of candidate feature subsets selected by PFA on different nodes of the tree with MAS dataset. The similar experiments for VIHASI, HOMELAB and combination MAS+VIHASI dataset can be see in Figure 9(b), Figure 9(d) and Figure 9(c).

For MAS dataset it can be noticed that for Node 4 the best fit is offered by 15-feature subset, and for Node 5 by 17 features. Also with VIHASI dataset Node 4 fits best with 15 features. For HOMELAB there is no better fit for any of the nodes than with the full 29 features. Notice that HOMELAB dataset does not contain running action, so there is no need to further classify in Node 2. With the combination dataset a slight improvement could be expected when using 17 features for Node 4. All in all, the expected gain when compared to using the full set of 29 features appears to be small. Overall, the performance of CRF as a temporal model does not decrease even when adding redundant features [7].

6. EXPERIMENTS: COMPARISON OF MULTI-VIEW SCENARIOS

All the previously introduced five camera scenarios were tested with all three datasets HOMELAB, MAS, VIHASI and combination dataset MAS+VIHASI. All the accuracy values represented here are averaged results of leave-one-
Figure 7: Feature correlation within three datasets, order starting from left being HOMELAB, MAS and VIHASI.

(a) HOMELAB

(b) MAS

(c) VIHASI

Figure 8: Feature correlation of entire (Node 1) dataset for PFA selected candidate feature subsets of 3,5,7,9,11,13,15,17, and 29 features. The indeces of selected features are shown in top of each sub-image.

6.1 Averaged results

For results on each dataset with each camera scenario see Table 4. With each of the experiments a minor improvement over SView scenario is already visible in the KView scenario. Even though the silhouette-based criteria for selecting the best view does not guarantee the most accurate classification of all the POVs, this approach performs better than keeping with a single view.

A greater improvement is achieved over all the datasets when using information from all the POVs. When considering the ideal silhouette datasets MAS and VIHASI the CView outperforms PViews, in average by 2.5%. With visibility and noise affected silhouettes of HOMELAB and with combination dataset having greater variability in the actions and provided views, it is PViews that performs better in average by 1.6%.

HOMELAB turned out to be the most difficult environment to classify in, best scenario PViews reaching accuracy of 69.1%. There are three major factors that cause this. First, as no ground truth silhouettes were available for the dataset, the silhouette extraction used could not provide full clean silhouettes. Secondly, people were not fully visible during the entire routine, as they were sometimes occluded by other objects in the scene such as tables and chairs. Also the carried objects, book and cup, deteriorate silhouette quality by including non-human pixels to foreground. Thirdly, sometimes people only partially fit in the field-of-view (FOV) of a camera. In Figure 6 one can see examples of encountered occlusions and FOV constraints. Without these real-world challenges of extracting clean silhouettes and being able to fully and without occlusions observe people, the accuracy of the HOMELAB dataset would also be in the range of values achieved by MAS and VIHASI.

6.2 Separation of activities

For more details on classification accuracy per activity one of top-2 confusion matrices for each of the datasets for a specified scenario are given in Table 5. With MAS greatest difficulties are in separating standing-interacting from standing. VIHASI has more confusion between walking and both standing activities. The combination dataset adds to this cases some confusion to running activity. With the er-
Figure 9: Fit of the trained 1D CRF-model of different featuresets for each of the tree nodes for each of the datasets in terms of training error.

Figure 9: Fit of the trained 1D CRF-model of different featuresets for each of the tree nodes for each of the datasets in terms of training error.

7. DISCUSSION

This paper has described a silhouette-based human activity recognition system that relies on 3 different categories of image features and temporal modeling, in our case CRF-model, to distinguish between 5 to 6 different human activities. The focus of the current paper was to compare different ways of combining data within a local distributed vision network. Five basic camera use scenarios were introduced: SView, KView, OView, PViews and CView. These scenarios were expected to represent good starting points for further development and optimization for multi-view data handling within the context of temporal activity modeling.

Experimental results demonstrate that good classification accuracy, above 91%, can be achieved by the proposed system when dealing with cleaner conditions provided by MAS and VIHASI datasets. With the combination dataset MAS + VIHASI still an accuracy above 80% was achievable. With the more challenging dataset of HOMELAB accuracy of 69% could still be reached, while few important real world problems arose. To continuously provide ideal silhouettes of people is in practice impossible, person is not always fully in the FOV of the cameras and occlusions with other objects exist. Whatever the feature representation, there will also exist variability within the same activity class. Variability is due to differences in appearance and pace of action of different people, and changing relations between observed people and the observing camera system.

For dealing with the inherent variability of features we examined the five proposed camera use scenarios. SView scenario was used as reference of a single-camera system performance. KView presented an attempt on how a three-part silhouette-based criteria in selecting a single view out of many views would increase performance. KView did offer 1-2% better performance than SView, but much more could still be achieved even by single-camera selection as the numbers with ideal OView show, reaching accuracy from 93 to 99%.

Next, two fusion scenarios CView and PViews were introduced. The difference between the scenarios was in which step of the process the data was fused. CView combined multiple feature streams into a single stream, whereas PViews combined label information from activity classification performed individually for each view. CView scenario offered best, most consistent accuracy, 91.4-92.2% with environments that have a similar setup for all cameras and that have people that behave in a similar manner. With environments with more variability, like HOMELAB and MAS+VIHASI, it is PViews which slightly outperforms CView scenario.
This would imply that fusion in the feature-level (CView) outperforms fusion in the label-level (PView) within more controlled settings. The more uncontrolled the environment is, the more likely better accuracy in multi-view activity analysis can be achieved by combining candidate labels.

8. REFERENCES