Fusion of Human Posture Features for Continuous Action Recognition*

Khai Tran, Ioannis A. Kakadiaris, and Shishir K. Shah

University of Houston, Department of Computer Science,
4800 Calhoun Road, Houston, TX 77204, USA
khaitran@cs.uh.edu, ioannisk@uh.edu, sshah@central.uh.edu

Abstract. This paper presents a real-time or online system for continuous recognition of human actions. The system recognizes actions such as *walking*, *bending*, *jumping*, *waving*, and *falling* and relies on spatial features computed to characterize human posture. The paper evaluates the utility of these features based on its joint or independent treatment within the context of the Hidden Markov Model (HMM) framework. A baseline approach wherein disparate spatial features are treated as an input vector to trained HMMs is used to compare three different independent feature models. In addition, an action transition constraints is introduced to stabilize the developed models and allow for continuity in recognized actions. The system is evaluated across a dataset of videos and results reported in terms of frame error rate, the frame delay in recognizing an action, action recognition rate, and the missed and false recognition rates. Experimental results shows the effectiveness of the proposed treatment of input features and the corresponding HMM formulations.

Keywords: Contiuous Action Recognition, HMMs, Fusion of Features.

1 Introduction

Recognizing human actions is a challenging problem that has received considerable attention from the computer vision community in recent years. This is especially the case due to its importance in various applications in the fields of surveillance and activity monitoring, human computer interaction, intelligent environments, etc. Each of these applications are domain specific and have additional requirements, but the general need for algorithms capable of detecting and recognizing human actions in real time remains fundamental. Broadly speaking, two primary considerations in analyzing human motion has been in modeling the temporal and spatial variations exhibited due to differences in duration of different actions performed and changing spatial characteristics of the human form in performing each action [1].

* This work is supported by, or in part by, the Norman Hackerman Advanced Research Program grant number 0003652-0199-2007.

K.N. Kutulakos (Ed.): ECCV 2010 Workshops, Part I, LNCS 6553, pp. 244–257, 2012.

Over the past few years, a number of approaches have been proposed to address these issues [2, 3]. Temporal templates have been proposed and used to categorize actions [4]. Methods that explicitly model relative changes in spatial descriptors over time [5], or estimates of global and local motion [6, 7] have also been utilized. More recently, spatio-temporal feature based approaches have been proposed and demonstrated for various action recognition applications. Representations based on a set of interest points are used to capture key variations in both space and time [8–12]. Space-time volumes built based on a global shape estimated by the human silhouette was proposed by Blank et al. [13]. Correlation of local features [14] and autocorrelation of features in space-time [15] have also been used to describe human movements.

In general, spatial variations can be effectively modeled by local appearance and shape representations while temporal variations require a more global treatment of the input image sequence. On the other hand, temporal variations can be effectively addressed by dynamic systems such as Hidden Markov Models (HMM) [1, 6]. With this capability, the burden of modeling spatial variations shifts to the selection of suitable feature sets to effectively discriminate different actions [16].

In this paper, we focus on the treatment of spatial features in the context of HMMs, specifically for the continuous recognition of actions within a real-time or online constraint. We outline the baseline approach wherein disparate spatial features are treated as an input vector to each HMM trained to recognize one of many actions. In addition, we impose an action transition constraint that explicitly accounts for the recognized action at the previous time step in recognizing an action at the current time step. Similar methods that rely on feature attributes have been proposed, many of which suffer from challenges in modeling the high-dimensional feature space [16]. We evaluate the baseline approach against a more independent treatment of the input features. Each feature is used to train HMMs for the specific action to be recognized and the output is fused to generate a final decision. Further, we also consider a weighted fusion model. The different treatments of input features and fusion methods are compared and evaluated for the problem of recognizing human action in an indoor environment, where the motions to be recognized include *'walking'*, *'bending'*, *'jumping'*, *'waving'* and *'falling'*. We propose the use of three features that describe human posture and evaluate the different models on a dataset of 17 videos, totaling over 7200 frames. The overview of the implemented system is shown in figure 1. The rest of the paper is organized as follows. Section 2 describes the human posture features proposed and used in this work. Section 3 describes the baseline model for action recognition as well as presents the different decision fusion models. Our experimental results and evaluations are presented in section 4, and section 5 concludes the paper.

2 Human Posture Features

Human posture features are primarily useful in describing actions representative of whole body movements. In this paper, we limit the set of actions to

Fig. 1. Schematic of the Human Action Recognition System

be recognized to common events in an indoor environment including *'walking'*, *'bending'*, *'jumping'*, *'waving'* and *'falling'*, all of which can intuitively be described by features that represent the human as a single object. One of the most common feature used is the height to width ratio (F_{HWR}) of a bounding box enclosing the detected region in the image [17, 18]. While this feature has been demonstrated to have good applicability, stable extraction of this metric can be challenging in real environments. In addition, every feature will be meritable for a limited number of actions and each action may require multiple features to be recognized robustly. Here, we propose two new features in addition to F_{HWR} that characterizes human posture.

Human actions can be described by combination of descriptors. In this paper, we concentrate more on human posture descriptors extracted from video frames. We propose two new descriptors that characterize human posture and we evaluate each proposed descriptors independently using our framework.

2.1 Nominal Height Ratio

The vertical height of the detected human in any image can provide an indication of the posture. In fact, as an absolute metric, such a feature can be used to identify progressive changes that may occur while a person performs actions such as *'standing'*, *'sitting'*, *'bending'*, and *'falling'*. Video metrology can provide beneficial information in ascertaining such an absolute metric [19]. We propose to compute the ratio of the true vertical height of the detected subject to the vertical height measured in the 2D image as a new feature, termed the nominal height ratio, F_{NHR}.

For estimating the true height of the detected person, consider a reference height given by the line segment A_0B_0, shown in figure 2. This vertical segment can be defined based on the first occurrence of the detected person blob in a video. The goal is to estimate the height of the same person when the person moves from position B_0 to B_1 on the ground. Using projective geometry, we determine the vertical vanishing point and horizontal vanishing line in the imaged scene [20]. The line segment A_0B_0 formed by the person standing at position B_0 on the ground will be parallel to line segment A_1B_1 formed by the person standing at position B_1 on the ground. By extension, line segment A_0A_1 and B_0B_1 will also be parallel. Under camera projection, line segment A_0B_0 and A_1B_1 will intersect at the vertical vanishing point Y and line segment A_0A_1 and B_0B_1 will intersect at point H on the horizontal vanishing line. Using the reference height, whenever we detect the point B_1 on the ground, we can determine the point A_1 by simple arithmetic calculations, which gives us the person's true height A_1B_1, up to a scale. To calculate the nominal height ratio feature, we use the bounding

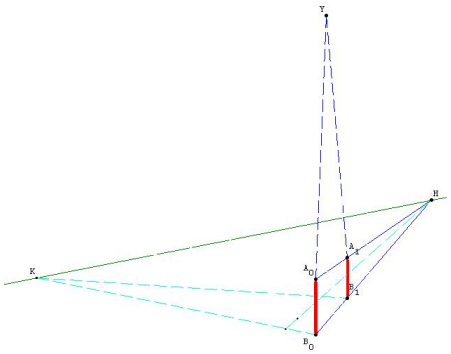

Fig. 2. Estimation of true height based on the estimated vanishing geometry

box around the detected person at position B_1. F_{NHR} is simply given by the the ratio of bounding box height and the person's estimated true height.

2.2 Upright Pose Model Projection Error

The second feature proposed in this paper is based on estimation of the 3D pose of the detected person in the 2D image. We use the POSIT algorithm [21] to estimate the pose of the detected person in the 2D image. For each detected region that bounds the person, we determine four 2D points that correspond to mid-points of the edges of the bounding box. We assume a simple cuboid as the 3D model representative of the person in a standing position. Next, we estimate a correspondence of the 2D and 3D points by aligning the cuboid to the 2D point along the bottom edge of the bounding box in the image. Back projecting rest of the 3D points of the cuboid allows us to estimate the position of the cuboid in the 2D image.

Let $p = \{p_1, ..., p_k\}$ be the set of k points within the bounding box in the 2D image. Further, let $P = \{P_1, ..., P_k\}$ be the set of corresponding points in the 3D model. Based on the POSIT algorithm [21], the projection of the 3D points in the image can be given as $\hat{p} = \{\hat{p}_1, ..., \hat{p}_k\}$. This allows us to now compute the error between the projection of the cuboid and the actual bounding box in the 2D image as the sum of distances between the original points and back-projected points. The estimated error is normalized by the size of bounding box. Hence, the upright pose model projection error, F_{UME}, can be given by:

$$F_{UME} = \frac{1}{B_h \times B_w} \sum_{i=1}^{k} (p_i - \hat{p}_i)^2, \tag{1}$$

where B_h and B_w are the height and width of the bounding box in the 2D image, respectively.

2.3 Height-to-Width Ratio

Height-to-width ratio, F_{HWR}, is a commonly used feature to recognize human actions [17, 18]. This feature is easy to compute once the human silhouette has been detected in an image. It is simply given by the ratio of the height to width of the bounding box enclosing the human silhouette. Nonetheless, since the feature is estimated entirely based on the 2D image information, it is susceptible to errors due to different body types such as tall, short, wide, etc.

Figure 3 shows a plot of the three features for a representative set of images taken from a video sequence. Each of the feature is a continuous valued signal. As can be seen, when a person is *'walking'*, *'jumping'* or *'waving'*, F_{NHR} and F_{HWR} has a higher value while F_{UME} has a lower value. This is quite the opposite when a person is *'bending'* or *'falling'*. Moreover, visual inspection suggests that the F_{NHR} is a more stable feature than other two features for each of the actions. The appropriate treatment of each of these features and their suitability for the action recognition task at hand is based on the models described in the next section.

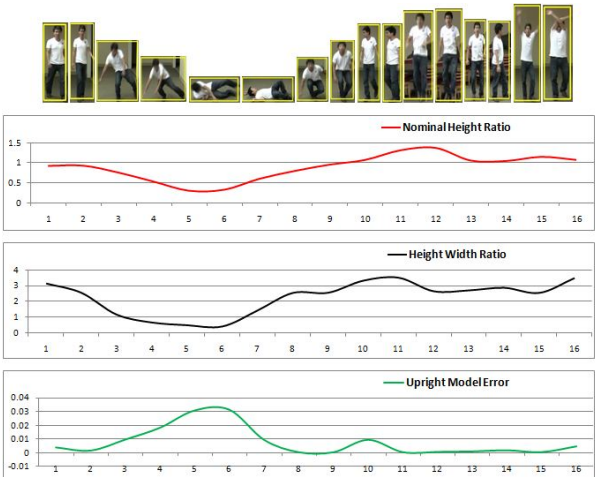

Fig. 3. Visualization for three human posture features calculated from representative video frames

3 Action Recognition

Hidden Markov Model is one of the parametric approaches that can model time-sequential data, provide time-scale variability, and is effective in learning a particular temporal problem. HMMs have proved useful in many fields, specially in speech recognition [22]. For a detailed explanation of HMMs, we refer the reader to the excellent tutorial by Rabiner [23]. HMMs have also been used in recognition of complex human actions that don't have accurate start and end times.

As an example, *'bending'* and *'falling'* actions can occur over different time scales wherein *'bending'* is potentially an action that occurs over a longer duration as compared to *'falling'* action. The first work that used Hidden Markov Model for human action recognition was proposed by Yamato et al. [24]. They successfully used the HMM with discrete observation symbols to model and recognize tennis shots.

Broadly speaking, there exists two variations of HMM implementations based on the nature of observation symbols, either discrete or continuous. Discrete HMMs require quantization or conversion of a continuous valued input signal to discrete observation symbols. Many approaches have been proposed to do so, including supervised and unsupervised mechanisms such as classification and clustering. In each case, this imposes additional assumptions on the properties of the input signal or features. Further, the generation of a comprehensive discrete vocabulary can be a very challenging problem. On the other hand, continuous HMMs can directly model a continuous valued input signal, although the learning process is much more complicated. Since we are interested in recognizing continuous actions and the features used in this work are continuous valued signals, we opt to build HMMs with continuous observation symbols and the input signal is modeled by a mixture of Gaussians.

In evaluating the human posture features within the HMM framework, we describe four formulations, each providing a different treatment of the input features. In the first model, the posture features are considered conditionally dependent and hence are modeled according to their joint probability distribution. In each of the other three models, the features are considered to be conditionally independent allowing for the design of simpler models for recognizing each action, wherein the contribution of each feature is realized by the fusion of the distinct simpler models.

In presenting the models, let us define the following notations. Let $E = \{e_1, ..., e_M\}$ be the set of M actions to be recognized, $V = \{v_1, ..., v_t\}$ be the frames of the input video of length t, $o_t = [F_t^1, ..., F_t^N]^T$ be the N features extracted from frame v_t, and $O = \{o_1, ...o_t\}$ be the sequence of observation symbols for the input video.

In general, the Hidden Markov Model can be defined as $\lambda = \{A, B, \pi\}$, where, A is the state transition probability distribution, B is the observation symbol's probability distribution for a given state, and π is the initial state distribution. For each action $e \in E$, we build an HMM λ^e and estimate the model parameters (A, B, π) that optimize the likelihood of the corresponding training observations. With these trained parameters, for a sequence of video frames V with unknown actions to be recognized, we extract posture features O and estimate the likelihood of the observation belonging to action e. Using the likelihood, $P(O|\lambda^e)$, estimated across all HMMs for each possible action $e \in E$, we select the action with the highest probability, given as:

$$e^* = \arg\max_{e \in E} P(O|\lambda^e) \tag{2}$$

The Baum-Welch algorithm [23, 25] is used to train the HMM for each action and estimate the optimal model parameters. During HMM training phase, parameter initialization is a very important stage that affects the model performance. Initial parameters estimated would be ideal if the the local and global maxima of the likelihood function computed would be the same. Unfortunately, there is no simple way for initialization of HMM parameters, and randomization does not always give the optimal solution in training HMMs. In our work, we use a preprocessing step that manually segments the observation into states to get a good initial estimate of model parameters. The observation itself is modeled by a mixture of Gaussians to parameterize the observation symbol distribution. The probability density function of sequence observation O is given as:

$$p(O) = \sum_{c=1}^{C} w_c \mathcal{N}(\mu_c, \Sigma_c) \tag{3}$$

where $\mathcal{N}$ is the normal distribution with mean μ_c and covariance Σ_c, C is the number of Gaussians, and w_c are the mixing weights for each Gaussian. For action recognition task, the Viterbi algorithm [23] is used to estimate the best state sequence given observation. Each action model is scored for a given observation and the action with the highest probability is selected as the recognized action.

3.1 Continuous Action Constraint

In a continuous action recognition task, the transition between actions is often constrained. Consider the example of a person walking and tripping, resulting in a fall. One logical evaluation of this event would result in actions *'walking'*, followed by *'bending'*, and *'falling'*. It would not be possible to observe *'falling'* without first observing *'bending'*. Hence, *'bending'* constitutes a transition constraint between *'walking'* and *'falling'*. Figure 4 pictorially represents this transition constraint among actions to be recognized. To effectively incorporate this constraint, we propose a modification to the traditional HMM framework, similar to grammar constraint in connected word recognition [23].

3.2 Posture Features with Continuous Action Constraint: A Baseline Model

A critical issue in implementing HMMs is the choice of how the features are being modeled. The features can be combined as one feature vector modeled as a mixture of Gaussians or the features can be modeled separately leading to multiple HMM models. In the latter case, the output of multiple HMMs has to be fused to recognize one action. In this subsection, we outline our baseline design that combines all features as one vector and integrates the model with Continuous Action Constraint, as described in figure 5. For simplicity, we will refer to this design as HMM_{Base}. The Continuous Action Constraint is incorporated by simply adding a binary weight vector $\overrightarrow{\alpha}$ of dimension M for the M HMM action models. Given the observation O, using Viterbi algorithm we find

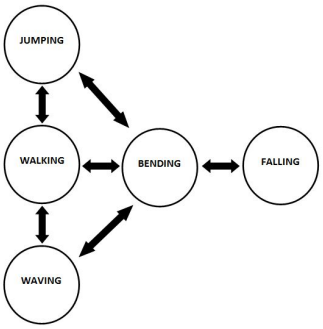

Fig. 4. Continuous Action Constraint

the optimal state sequence. Then, we compute probability of observation given each model $P(O|\lambda^e)$. The action at time t is decided by selecting the maximum weighted probability:

$$e_t^* = \arg\max_{e \in E} \left[\alpha_t^e P(O|\lambda^e)\right] , \tag{4}$$

where α_t^e is the weight of HMM model for action e at time t and is controlled by:

$$\begin{aligned} \alpha_t^e &= 1 \text{ if } e_{t-1}^* \text{ and } e \text{ are in order} \\ \alpha_t^e &= 0 \text{ if } e_{t-1}^* \text{ and } e \text{ are not in order} \end{aligned} \tag{5}$$

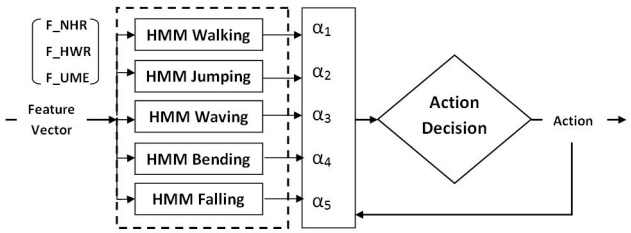

Fig. 5. Baseline HMM model with Continuous Action Constraint

3.3 Fusion Models

Joint modeling of high-dimensional feature vectors can often result in higher model fitting errors [16]. Hence, reducing feature dimensions by modeling them as independent observations can decrease the fitting error and may enhance performance of the HMMs. It is common to observe that certain features are good to describe some actions while others are not, and we may not effectively leverage

or learn this information if we combine all the features. For these reasons, we consider a fusion of models framework that treats each feature independently and a fusion module is used to combine the resulting output probabilities for a final recognition of action.

Consider a set of N features $F = F^1, ..., F^N$. With actions we have denoted above, lets define HMMs $\lambda_{F^i}^{e_i}$ to model human action $e_i \in E$ for each feature $F^i, 1 \leq i \leq N$. As a result, we will have $M \times N$ HMMs for the M actions. The framework integrated with Continuous Constraint Action is as shown in figure 6. For each feature, using the Viterbi algorithm, we can calculate the probability

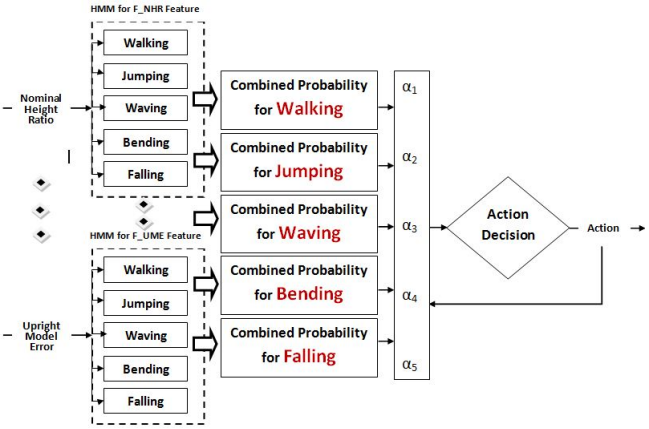

Fig. 6. Multiple HMM models within a fusion framework with Continuous Action Constraint

of single feature observation, $P(O|\lambda_{F^i}^{e_i})$, given by each model. Now with $M \times N$ normalized probability outputs, we propose four fusion mechanisms to obtain a final decision; the first based on the sum rule, the second based on the product rule, the third based on the weighted sum rule, and the fourth based on the weighted product rule [26], resulting in models that we denote as HMM_{Sum}, HMM_{Prod}, and HMM_{WSum}, and HMM_{WProd}, respectively.

Sum Rule: For the decision based on the sum rule, we compute the sum of all probabilities as $\sum_{i=1}^{N} P(O|\lambda_{F^i}^{e_j}), 1 \leq j \leq M, 1 \leq i \leq N$, and the action recognized is given by:

$$e_t^* = \arg\max_{e \in E} \ [\alpha_t^e \sum_{i=1}^{N} P(O|\lambda_{F^i}^{e})] \ , \tag{6}$$

where α_t^e is the weight of HMM model for action e at time t, controlled by the Continuous Action Constraint.

Product Rule: Similarly, for decision based on the product rule, we compute all product of probabilities as $\prod_{i=1}^{N} P(O|\lambda_{F^i}^{e_j}), 1 \leq j \leq M, 1 \leq i \leq N$, and select the action that has maximum product probability as:

$$e_t^* = \underset{e \in E}{\arg\max} \ [\alpha_t^e \prod_{i=1}^{N} P(O|\lambda_{Fi}^e)]. \tag{7}$$

Weighted Sum Rule: While the fusion of probabilities computed by each HMM can be beneficial, it does not allow one to evaluate the merit of each contribution. This can easily be incorporated by introduction of weights associated with each HMM for each action. Hence, the fusion model can now be formulated according to the weighted sum or product rule [26].

Let $\Phi = \{\phi_1^e, ..., \phi_N^e\}$ be the set of weights associated with each of the N HMMs for each action e. For the decision based on weighted integration, we compute the sum of all weighted probabilities as $\sum_{i=1}^{N} \phi_i^{e_j} P(O|\lambda_{F^i}^{e_j}), 1 \leq j \leq M, 1 \leq i \leq N$, and the action recognized is given by:

$$e_t^* = \underset{e \in E}{\arg\max} \ [\alpha_t^e \sum_{i=1}^{N} \phi_i^e P(O|\lambda_{F^i}^e)]. \tag{8}$$

Weighted Product Rule: Similarly, for the weighted product model, the action is recognized based on computing the product of exponential weighted probabilities as $\prod_{i=1}^{N} P(O|\lambda_{F^i}^{e_j})^{\phi_i^{e_j}}, 1 \leq j \leq M, 1 \leq i \leq N$. Hence, the final action recognized is given by:

$$e_t^* = \underset{e \in E}{\arg\max} \ [\alpha_t^e \prod_{i=1}^{N} P(O|\lambda_{F^i}^e)^{\phi_i^e}]. \tag{9}$$

4 Experiments and Results

In evaluating the human posture features within the implementation of different HMM formulations, we address the task of recognizing five actions, *'walking'*, *'bending'*, *'jumping'*, *'waving'*, and *'falling'* on real data sets acquired within an indoor environment. A total of 17 videos (over 7200 frames) are including in our dataset. 3 videos are used for the purpose of training the parameters of the different HMM models for each action. In addition, 3 other videos are used for validating the models and estimating the contribution (weights ϕ) for each HMM model set for the weighted sum formulation HMM_{WSum}, and the weighted product formulation HMM_{WProd}. The remaining 11 videos are used for testing. We use the mixture of Gaussians model for learning the background and detecting the foreground blobs. Detected blobs are morphologically processed to establish a bounding box. A Kalman filter based tracker is used track the blob where the tracker uses the blob centroid, and the height and width of the bounding box as the state variables. To establish ground-truth, each frame in all videos is manually annotated.

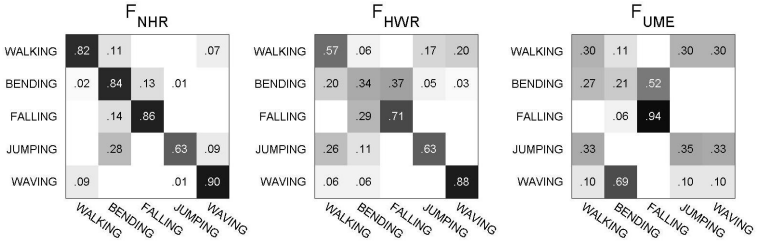

Fig. 7. Confusion matrix showing the accuracy at frame level of recognized actions for individual posture features

One of the key parameters in the design of an HMM is the length of the observation symbols. For a continuous action recognition problem, this cannot be known *a priori*, we empirically estimate this based on the training and validation videos. During training, we vary the length of observations from t_{min} to t_{max} for all HMM models simultaneously and pick the optimal observation length as the one that results in the most accurate recognition of actions in the validation videos. The benefit of a large window size is that the action to be recognized will be modeled more accurately, resulting in a higher recognition accuracy. The limitation on the other hand is that this will result in an increase in the delay between the actual action performed and its recognition by the system.

Among the fusion models evaluated in this paper, the weighted models require that each set of HMMs trained for a single posture feature be evaluated for its contribution towards the final action recognition. To do so, we evaluate the accuracy of the set of HMMs per feature on the validation videos. Accuracy is measured by comparing the recognized action for each frame of the video against the manually annotated ground-truth. Figure 7 shows the computed accuracy for HMM models that use a single feature as input. The weights used for the weighted models, HMM_{WSum} and HMM_{WProd}, are simply the accuracy of correctly recognizing each action for each of the three independent feature models.

4.1 Analysis of Features and Models

Having estimated the parameters for each of the different models, the accuracy of recognizing actions was computed based on testing across the 11 testing videos. Table 1 summarizes the results and figure 8 shows the confusion matrix for each of the models. Accuracy was calculated in terms of the frame error rate, the frame delay in recognizing an action, action recognition rate, and the missed and false recognition rates. Frame error rate was calculated by simply measuring the number of false recognized actions per frame. On the other hand, action recognition rate is reported based on the actual number of action transitions detected against the number present in the input video. The best action recognition rate obtained by a single feature model is 91.91%, one based on the

Table 1. Results indicating accuracy of the four different models across the testing dataset

Model	Frame Err. Rate	Frame Delay(s)	Action Recog. Rate	Missed/False Rate
$HMM_{F_{NHR}}$	17.24%	5.2	91.91%	1.57%
$HMM_{F_{HWR}}$	41.64%	6.31	82.41%	2.78%
$HMM_{F_{UME}}$	42.66%	5.11	85.84%	2.43%
HMM_{Base}	29.63%	3.67	91.91%	1.04%
HMM_{Sum}	21.39%	3.43	97.13%	0.70%
HMM_{Prod}	17.14%	3.50	98.03%	0.61%
HMM_{WSum}	16.91%	3.24	99.07%	0.52%
HMM_{WProd}	16.43%	3.17	99.22%	0.35%

nominal height ratio. While all the three features perform reasonably well and show some correlation, there is still sufficient variability as seen based on their frame error rates that is effectively exploited by the fusion models. As a result, the the fusion models outperform any individual feature and the base model that simply uses them as a vector of features performs at par with the best feature model. The sum and the product model show an increase in the recognition rate as well as a drop in the frame error rates suggesting their ability to appropriately exploit the diversity in the features. The weighted models account for errors associated with continuous recognition at each frame and show a corresponding increase in the overall action recognition rate. The best performing model is the weighted product model that shows 99.22% recognition rate with a frame error rate of 16.43%. In general, the Viterbi algorithm requires the full observation

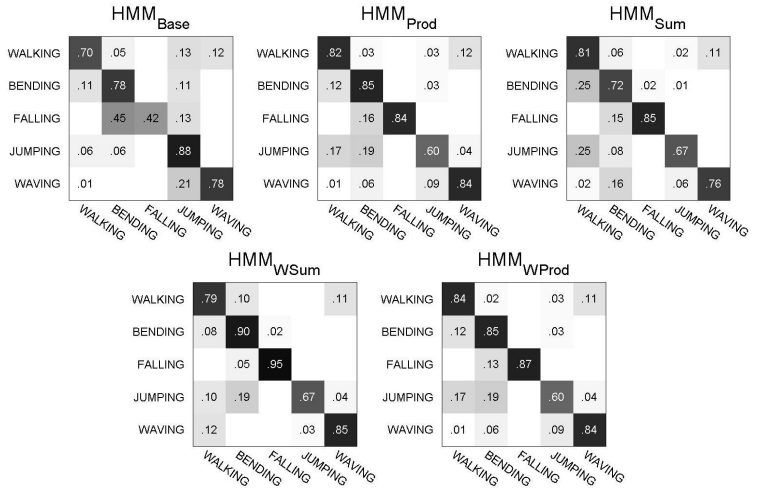

Fig. 8. Confusion matrices showing the frame level action recognition accuracy for the different HMM models

sequence to calculate the probability of observation given the model before backtracking to resolve the complete sequence. During training, we optimize for this delay and the results on the testing videos show that the recognition delay is, on average, 3 to 5 frames across all models. For an input video at 30 frames/second, this amounts in a delay of $\approx$ 0.1-0.2 seconds.

5 Conclusion

In this paper, we have presented a system for the continuous recognition of human actions and evaluated the same for a set of actions in an indoor environment. We have proposed human posture features and focused on their treatment in the context of Hidden Markov Models, specifically for the recognition of actions within a real-time or online constraint. We use a baseline approach wherein disparate spatial features are treated as an input vector to each HMM trained to recognize one of many actions. We evaluate the baseline approach against a more independent treatment of the input features. The different treatments of input features and fusion methods are compared and evaluated. Overall, the obtained results demonstrate the benefit of each of the human posture features and, more importantly, the gain of treating the features as independent observations, thereby maximizing the redundancy and diversity of information. The result also shows that the independent feature models, HMM_{Sum}, HMM_{Prod}, HMM_{WSum}, and HMM_{WProd} are more stable and accurate. The fusion models HMM_{Sum} and HMM_{Prod} clearly outperform HMM_{Base}.

References

1. Kellokumpu, V., Pietikäinen, M., Heikkilä, J.: Human activity recognition using sequences of postures. In: Machine Vision Applications, pp. 570–573 (2005)
2. Gavrila, D.M.: The visual analysis of human movement: a survey. Comput. Vis. Image Underst. 73, 82–98 (1999)
3. Moeslund, T.B., Hilton, A., Krüger, V.: A survey of advances in vision-based human motion capture and analysis. Comput. Vis. Image Underst. 104, 90–126 (2006)
4. Bobick, A.F., Davis, J.W.: The recognition of human movement using temporal templates. IEEE Transactions on Pattern Analysis and Machine Intelligence 23, 257–267 (2001)
5. Ali, A., Aggarwal, J.: Segmentation and recognition of continuous human activity. In: Proceedings of IEEE Workshop on Detection and Recognition of Events in Video, pp. 28–35 (2001)
6. Eickeler, S., Kosmala, A., Rigoll, G.: Hidden markov model based continuous online gesture recognition. In: ICPR 1998: Proceedings of the 14th International Conference on Pattern Recognition, vol. 2, p. 1206. IEEE Computer Society, Washington, DC (1998)
7. Iwai, Y., Shimizu, H., Yachida, M.: Real-time context-based gesture recognition using hmm and automaton. In: IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real -Time Systems, p. 127 (1999)
8. Laptev, I.: On space-time interest points. Int. J. Comput. Vision 64, 107–123 (2005)

9. Dollár, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior recognition via sparse spatio-temporal features. In: VS-PETS (2005)
10. Niebles, J.C., Fei-Fei, L.: A hierarchical model of shape and appearance for human action classification. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2007)
11. Niebles, J.C., Wang, H., Fei-Fei, L.: Unsupervised learning of human action categories using spatial-temporal words. International Journal of Computer Vision (2008)
12. Sun, X., Chen, M., Hauptmann, A.: Action recognition via local descriptors and holistic features. In: Computer Vision and Pattern Recognition Workshop, pp. 58–65 (2009)
13. Blank, M., Gorelick, L., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. In: ICCV 2005: Proceedings of the Tenth IEEE International Conference on Computer Vision, pp. 1395–1402. IEEE Computer Society, Washington, DC (2005)
14. Shechtman, E., Irani, M.: Space-time behavior based correlation. In: CVPR 2005: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, vol. 1, pp. 405–412. IEEE Computer Society, Washington, DC (2005)
15. Kobayashi, T., Otsu, N.: Action and simultaneous multiple-person identification using cubic higher-order local auto-correlation. In: ICPR 2004: Proceedings of the 17th International Conference on Pattern Recognition, ICPR 2004, vol. 4, pp. 741–744. IEEE Computer Society, Washington, DC (2004)
16. Ribeiro, P.C., Santos-victor, J.: Human activity recognition from video: modeling, feature selection and classification architecture. In: International Workshop on Human Activity Recognition and Modeling, HAREM (2005)
17. Miaou, S.G., Sung, P.H., Huang, C.Y.: A customized human fall detection system using omni-camera images and personal information. In: 1st Transdisciplinary Conference on Distributed Diagnosis and Home Healthcare, pp. 39–42 (2006)
18. Anderson, D., Keller, J.M., Skubic, M., Chen, X., He, Z.: Recognizing falls from silhouettes. In: 28th EMBC International Conference on Engineering in Medicine and Biology Society, pp. 6388–6391 (2006)
19. Guo, F., Chellappa, R.: Video metrology using a single camera. IEEE Transactions on Pattern Analysis and Machine Intelligence 32, 1329–1335 (2010)
20. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press (2000) ISBN: 0521623049
21. DeMenthon, D., Davis, L.S.: Model-based object pose in 25 lines of code. International Journal of Computer Vision 15(1-2) (1995)
22. Rabiner, L., Juang, B.: An introduction to hidden markov models. IEEE ASSP Magazine 3, 4–16 (1986)
23. Rabiner, L.R.: A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of the IEEE, 257–286 (1989)
24. Yamato, J., Ohya, J., Ishii, K.: Recognizing human action in time-sequential images using hidden markov model. In: Computer Vision and Pattern Recognition, pp. 379–385 (1992)
25. Baum, L.E., Petrie, T., Soules, G., Weiss, N.: A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. Ann. Math. Statist., 164–171 (1970)
26. Kittler, J.: Combining classifiers: A theoretical framework. Pattern Analysis and Applications 1, 18–27 (1998)