

Online, Real-time Tracking and Recognition of Human Actions *

Pradeep Natarajan, Ramakant Nevatia
Institute for Robotics and Intelligent Systems,
University of Southern California,
Los Angeles, CA 90089-0273
{pnataraj, nevatia}@usc.edu

Abstract

We present a top-down approach to simultaneously track and recognize articulated full-body human motion using learned action models that is robust to variations in style, lighting, background, occlusion and viewpoint. To this end, we introduce the Hierarchical Variable Transition Hidden Markov Model (HVT-HMM) that is a three-layered extension of the Variable Transition Hidden Markov Model (VTHMM). The top-most layer of the HVT-HMM represents the composite actions and contains a single Markov chain, the middle layer represents the primitive actions which are modeled using a VTHMM whose state transition probability varies with time and the bottom-most layer represents the body pose transitions using a HMM. We represent the pose using a 23D body model and present efficient learning and decoding algorithms for HVT-HMM. Further, in classical Viterbi decoding the entire sequence must be seen before the state at any instant can be recognized and hence can potentially have large latency for long video sequences. In order to address this we use a variable window approach to decoding with very low latency. We demonstrate our methods first in a domain for recognizing two-handed gestures and then in a domain with actions involving articulated motion of the entire body. Our approach shows 90-100% action recognition in both domains and runs at real-time (≈ 30 fps) with very low average latency (≈ 2 frames).

1. Introduction

Our objective is to develop a system for automated human gesture and action recognition that takes monocular videos as input. Such capabilities are needed for a number of applications in human computer interaction, including assistive technologies and intelligent environments. To

*This research was supported, in part, by the Office of Naval Research under Contract N00014-06-1-0470 and, in part, by the VACE program of the U.S. Government. The views expressed here do not necessarily reflect the position or the policy of the United States Government.

function in interactive environments, such a system must be real-time and on-line (*i.e.* have low latency). Furthermore, it is desirable that the method be capable of being trained with only a small amount of training data.

The traditional approach to activity recognition is *bottom-up* where an actor is segmented from the background (usually by "background subtraction" methods), then the body and the 3-D pose are tracked and the trajectories are used for action/gesture recognition. Such an approach scales well with the number of possible activities and also can potentially handle previously unseen events. However, accurate tracking at the lower level is difficult since we use 23 degrees of freedom in our pose estimate, so the search space is huge. Further, variations in the style of actions, background and clothing of the actor, viewpoint and illuminations can introduce errors. Alternatively, one can take a *top-down* approach where higher level event models drive tracking; this could be considered similar to the *tracking-as-recognition* approach of [18]. While the complexity of this approach increases linearly with the number of actions, it is also more robust to low-level errors and is especially attractive when we are interested in recognizing a finite set of predefined actions like the ones we consider here. Hence we take a top-down approach to action recognition in our work.

Our basic model is a variation of the well-known Hidden Markov Model (HMM). However, rather than a single level model, we use a hierarchical model with three layers. Top level is for recognizing compositions of simpler events that we call *primitive* events; primitive events form the middle level and the bottom level is responsible for tracking the poses. Initial segmentation of the contour is provided by a background subtraction module.

HMMs have been a popular tool in recognition of activities in video and also in speech recognition. They map well to these problems and are relatively simple. However, the basic HMM has several deficiencies: a) A single variable state representation is insufficient for modeling multiple interacting agents/parts, b) there is no direct way to model the

inherent hierarchical structure of actions, and c) the implied duration models for state occupancies are exponentially decaying which is unrealistic.

Several extensions of HMMs to include more flexible duration models have been suggested, including the hidden semi-Markov models (HSMM) [7, 5] with explicit duration models and the variable transition HMMs (VT-HMMs) where the transition probability explicitly depends on the time duration of the current state [11] (originally called inhomogeneous HMM in [11]). This flexibility in duration models, however, comes at a cost in the complexity of the learning and decoding algorithms. We build on the VTHMM formalism, as the algorithms for it are much more efficient than for HSMM while flexibility of duration modeling is maintained. Thus, we call our multi-layer model to be a Hierarchical VT-HMM or HVT-HMM. Later in the paper, we present training and decoding algorithms for HVT-HMM and show that the training can be accomplished with a small training set and that decoding is highly efficient. Standard decoding algorithms for HMMs, such as the well-known Viterbi algorithm, require the complete sequence to be seen to find an optimal path; this creates unacceptable latencies for an interactive system. Instead, we develop an on-line method, with low latency; our method is an extension of an algorithm presented in [8].

The rest of the paper is organized as follows - In section 2 we briefly describe related work and how our approach compares to them, in section 3 we define HVT-HMM formally, derive the learning and decoding algorithms and also describe the body model and observation functions, in section 4 we present experimental results and finally present conclusions and future work in section 5.

2. Related Work

Many methods for activity and gesture recognition have been developed in the past. Most use some variation of HMMs for reasoning. There are three main variations. One is the use of multiple variables to represent states of different agents at each instant, to accommodate actions of multiple agents ([15, 3]). Second is use of hierarchical structures to model activities at different levels of abstraction ([10, 5]). Third is introduction of explicit models for state durations ([5, 7]). Recently, *discriminative* models like conditional random fields (CRF) are being increasingly used for gesture and action recognition [16, 14] as an alternative to the *generative* HMMs. While such discriminative approaches have shown promising results in several domains, they also typically require large amount of training data before the performance gains are achieved ([9]). Since in our work we aimed to minimize training requirements, we chose to use the generative approach.

A vast majority of existing methods for action recognition use a *bottom up* approach where objects/pose is first

tracked and then fed to the higher level recognition layers. [2] introduced motion-energy images (MEI) and worked by extracting key frames for each action during the training phase and then recognized actions in the test sequences by estimating the distance from the key frames. [17] generalized MEIs to 3D using cameras at multiple view-points. [10, 5, 7] also present systems where the objects of interest are first tracked and the tracks are then used to recognize activities. While these methods have been shown to perform well in various domains they require accurate tracking and are likely to fail in cases where occlusion or other factors introduce errors in the lower layers.

In recent work, [4] attempts to address the tracking errors by using a top-down approach with prior event models to match broken tracks (called tracklets). [18] uses prior event models with a discrete set of motion templates where at each instant the templates are matched with the foreground and the event sequence is recognized by doing a Viterbi search on the templates. Our method builds on this idea to do continuous pose tracking and uses the feedback from the lower layers for action recognition.

3. HVT-HMM

In this section, we present the theory of HVT-HMM in 3 steps - 1) we present a formal definition and define the parameters of HVT-HMM. 2) Then we present our approach to parametrize actions with the HVT-HMM and learn the parameters, and finally 3) Develop an efficient decoding algorithm and extend it to online-decoding.

Notations: C = No. of composite events, P = Max no. of primitive events per composite event, D = Max duration that can be spent in a primitive, N = Max no. of poses possible under a primitive action, T = No. of frames in video.

3.1. Model Definition and Parameters

We begin by defining a standard HMM model λ by the tuple (Q, O, A, B, π) where, Q is the set of possible states, O is the set of observation symbols, A is the state transition probability matrix ($a_{ij} = P(q_{t+1} = j | q_t = i)$), B is the observation probability distribution ($b_j(k) = P(o_t = k | q_t = j)$) and π is the initial state distribution. It is straightforward to generalize this model to continuous (like gaussian) output models.

The hierarchical hidden Markov model (HHMM) extends this by including a hierarchy of hidden states. This can be formally specified by the tuples- $\lambda' = (Q_h, O_h, A_h, B_h, \pi_h)$ where $h \in 1..H$ indicates the hierarchy index.

In traditional HMMs, constant state transition probability (a_{ij}) and the first order Markov assumption implies that the duration probability of a state decays exponentially with time. Variable transition hidden Markov models (called

where,

- n is a suitable normalization constant
- p_t is the primitive event at time t
- $\mu_{p_{t-1}}$ denotes the average time spent in executing primitive action p_{t-1} and is learned from training data.
- $d_{p_{t-1}}$ denotes the time spent in primitive p_{t-1} in the current execution of the action.
- σ is a noise parameter to allow for some variation in the actual end position of the actions. We set this to 20% of $\mu_{p_{t-1}}$ in our implementation.

The intuition behind the choice of the sigmoid function is that it allows state transitions only when the primitive event has been executed for a duration close to the mean. This is because as $d_{p_{t-1}}$ approaches $\mu_{p_{t-1}}$, term 1 (for maintaining current state) in equation 2 decreases and term 2 (for transition to next state) increases as illustrated in Figure 3.

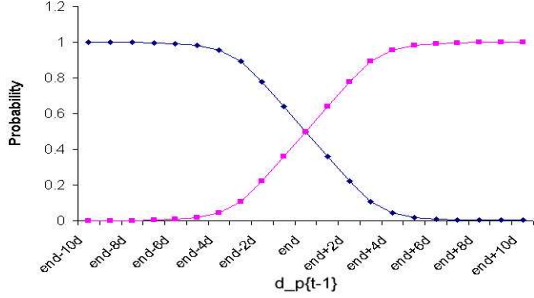


Figure 3. Variation in values of Equation 2 - Line with diamonds-term1, Line with squares-term2

At the **Track Layer** we define the transition probability $P(x_t|\{x_{t-1}, p_t\})$ as follows-

$$P(x_t|\{x_{t-1}, p_t\}) = \begin{cases} \frac{1}{2M_{p_t}} & \text{if } 0 \leq |x_t - x_{t-1}| \leq 2M_{p_t}, \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where,

- M_{p_t} is the mean distance the pose changes in one frame (average speed) under primitive p_t and is learned during training.

- x_t is obtained from x_{t-1} using a simple geometric transformation specified by p_t .

- $|x_t - x_{t-1}|$ is the distance between x_t and x_{t-1}

This definition basically restricts the speed of a pose transition to at most $2M_{p_t}$. Based on these observation and transition probabilities, we can re-estimate the parameters μ_i and M_i using a forward-backward algorithm similar to the Baum-Welch algorithm used in basic HMMs. Further, since the actions in our experiments have typical primitive duration(μ_i) and speed of action (M_i) across all instances, estimates from a single sample for each action is sufficient to train the models.

3.3. Decoding Algorithm

Next we develop a decoding algorithm for *continuous* recognition of composite events from an unsegmented video stream. Let $\delta_t(c, i, x_t, d)$ denote the probability the maximum path that the composite event c is in primitive event i and pose x_t at time t and has also spent a duration d in primitive i . By assuming a unique start primitive (indexed by 1) for each composite event the δ variables can be calculated using the following equations-

$$\begin{aligned} \delta_{t+1}(c, j, x_{t+1}, d+1) &= \\ \max_{x_t} \delta_t(c, j, x_t, d) a_{ij}^c(d) p^c(x_{t+1}|x_t, j) p(I_{t+1}|x_{t+1}), d > 0 \\ \delta_{t+1}(c, j, x_{t+1}, 1) &= \\ \max_{i, \tau, x_t} \delta_t(c, i, x_t, \tau) a_{ij}^c(\tau) p^c(x_{t+1}|x_t, j) p(I_{t+1}|x_{t+1}), j > 1 \\ \delta_{t+1}(c, 1, x_{t+1}, 1) &= \\ \max_{c', i, \tau, x_t} \delta_t(c', i, x_t, \tau) (\tau) p^c(x_{t+1}|x_t, 1) p(I_{t+1}|x_{t+1}) \\ \delta_1(c, j, x_1, 1) &= \pi_j^c \pi_{x_1, j}^c p(I_1|x_1) \end{aligned} \quad (4)$$

where,

- $a_{ij}^c(\tau)$ is the probability of transition from primitive i to j having spent time τ in state i , under complex event c
- $p^c(x_{t+1}|x_t, j)$ is the probability of transition to pose x_{t+1} from x_t under primitive j
- $p(I_{t+1}|x_{t+1})$ is the observation probability at the track layer.

By calculating the δ 's for each c, j, x_t, d we can get MAP path from $\max_{c, j, x_T, d} \delta_T(c, j, x_T, d)$. Further by storing the previous states for each c, j, x_t, d at each instant t we can retrace the Viterbi path through the HVT-HMM.

This algorithm has a complexity of $O(TCPDN^2(P + C))$ which can be very slow as the number of possible poses(N) is large. Hence as an approximation, we store only the top K configurations at each instant and also prune out configurations $\{c, j, x_t, d\}$ for which

$$\max_{c', j', x'_t, d'} \delta_t(c', j', x'_t, d') - \delta_t(c, j, x_t, d) > p \quad (5)$$

where p is a sufficiently small threshold. In this case, each max in the RHS of equation 4 takes only $O(K)$ and hence the whole algorithm takes $O(KCPNDT)$.

One crucial issue with the basic Viterbi algorithm is that the entire observation sequence must be seen before the state at any instant can be recognized. Thus even in cases where the algorithm runs at real time, the average latency(time taken to declare the state at a given instant) can be infinitely large. One simple solution is to use a fixed look ahead window, but the performance of this algorithm is poor. Instead we build on the *online variable-window* algorithm proposed recently [8] which is demonstrated to be better than other approaches. At each frame we calculate the following function-

$$f(t) = M(t) + \lambda(t - t_0) \quad (6)$$

where,

- t_0 is the last frame at which the composite, primitive and track states were output.

- $M(t)$ can be any suitable function which is large if the probability at t is concentrated around a single state and small if it is spread out.

- The term $(t - t_0)$ penalizes latency

- λ is a weight term that also functions as a normalization factor.

With this metric, large values for λ penalize latency while smaller values penalize errors. In our implementation we choose $M(t)$ to be $(1 - \max_{c,j,x_t,d} \delta_t(c, j, x_t, d))$ as it is the simplest. Alternate choices include the entropy of the state distribution but as our results show the simple choice performs satisfactorily. At each instant, if $M(t) < \lambda(t - t_0)$ we output the maximum probability state at t , and the states in the interval $[t_0, t)$ that lead up to it, and then repeat the procedure from $(t + 1)$. [8] shows that this algorithm is 2 - competitive¹ for this choice of $M(t)$ in HMMs and a similar reasoning applies here as well.

Summary: To summarize the inference procedure for recognizing the events given a video, we do the following - Let $\langle c_t, p_t, x_t \rangle$ be a possible $\langle \text{composite} - \text{event}, \text{primitive} - \text{event}, \text{pose} \rangle$ tuple at time t . At each frame we first extract the foreground blob from the video using background subtraction. Next, we take the top-K tuples from $(t - 1)$ and generate all possible tuples for time t based on the constraints of the HVT-HMM graphical structure. For each of these tuples we correct for the scale(H) based on the height of the blob, and the orientation (for actions like walk/run) based on the instantaneous direction of motion. We then project each of these poses on the blob and choose the top-K tuples for time t and repeat the cycle for the next frame. At the first frame we choose the top-K tuples simply based on the observation probability in equation 1. We simultaneously calculate the metric in equation 6 for the top-K tuples at each frame and declare the states if $M(t) < \lambda(t - t_0)$ as described before.

4. Experiments

We tested our method in two domains - 1) For recognizing fourteen arm gestures used in military signalling and 2) For tracking and recognizing 9 different actions involving articulated motion of the entire body. All run time results shown were obtained on 3GHz, Pentium IV, running C++ programs.

4.1. Gesture Tracking and Recognition

In the first experiment, we demonstrate our method on the gesture datasets used in [6, 13], for examples in the do-

¹The cost is no more than 2 times the cost of any other choice of t

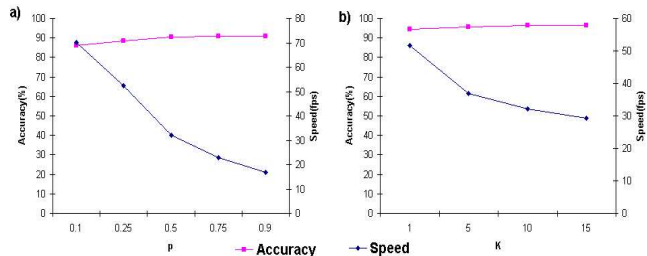


Figure 5. Variation in Accuracy(%) and Speed(fps) with - a) p b) K

main of military signaling which consists of fourteen arm gestures shown in Figure 4. The dataset consists of videos of the fourteen gestures performed by five different people with each person repeating each gesture five times for a total of 350 gestures.

Each gesture is modeled by a 3-state VTHMM at the primitive layer which in turn corresponds to 2 track layer HMMs (one for each hand). We trained the model for each gesture using a randomly selected sequence and tested the variation in performance of the decoding algorithm with the pruning parameters K and p . As can be seen from figure 5, while the accuracy increases slightly with larger K and p , the speed decreases significantly. Since our video is at 30fps, we choose $K = 10$ and $p = 0.5$ for real-time performance. Our system gives an overall accuracy of 90.6%. We also tested our method with large variations in background, lighting, individual style, noise, occlusion and also with significant variation in the actual angle with the camera and observed no drop in performance. Figure 6 illustrates some of these variations. Next we replaced the primitive event layer of HVT-HMM with a simple HMM (to give a HHMM) and the semi-Markov HSMM (to give a HS-HMM) instead of VTHMM and tested them under the same training and pruning conditions. As can be seen from the results in table 1 including duration models in HVT-HMM produces a significant improvement in performance over an HHMM without the drop in speed seen in HS-HMMs. Further, the average latency of HVT-HMM is lower than that of HHMM because duration dependent state transitions restrict the set of possible states and hence $M(t)$ in equation 6 tends to be large. Since no known online decoding algorithm exists for an HSMM, the entire sequence must be seen before the state sequence is generated for HS-HMM.

On the same dataset, [6] reports 85.3% accuracy on the first 6 gestures by testing and training on all 150 gestures (we have 97%). [13] reports 83.1% accuracy on the entire set. This approach uses edge contours instead of foreground silhouettes to allow for moving cameras, even though the data itself does not contain any such example; this may be partially responsible for its lower performance and hence a direct comparison with our results may not be meaningful.

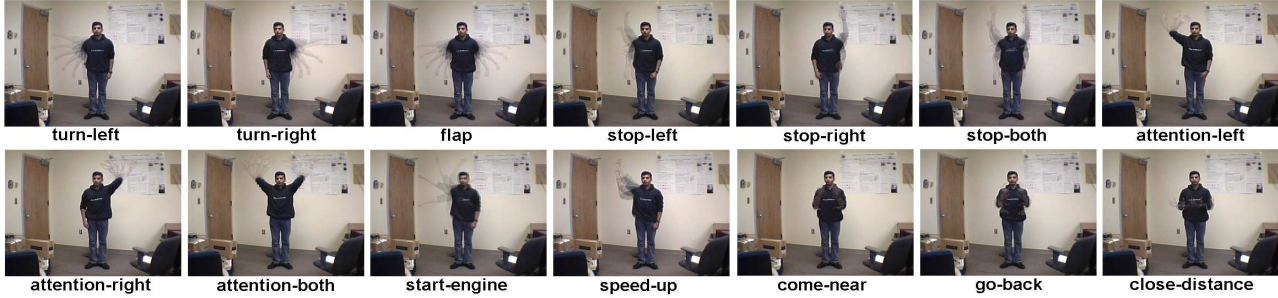


Figure 4. Overlay Images for Arm Gestures



Figure 6. Recognizing and tracking with variation in - a)Background and Lighting b) Style c) Random external motion d) Self-Occlusion

	Accuracy (%)	Speed (fps)	Avg. Latency (frames)	Max Latency (frames)
HVT-HMM	90.6	35.1	1.84	9
HHMM	78.2	37.4	4.76	14
HS-HMM	91.1	0.63	-	-

Table 1. Comparison of HVT-HMM, HHMM and HS-HMM Gesture Dataset

Another key feature of our method is that it requires just one training sample/gesture while [6, 13] require a *train : test* ratio of 4 : 1.

4.2. Tracking and Recognizing Articulated Body Motion

In the second experiment, we demonstrate our methods on a set of videos used in [1] of actions that involve articu-

	Accuracy (%)	Speed (fps)	Avg. Latency (frames)	Max Latency (frames)
HVT-HMM	100.0	28.6	3.2	13
HHMM	91.7	32.1	7.6	25
HS-HMM	100.0	0.54	-	-

Table 2. Comparison of HVT-HMM, HS-HMM and HHMM Action Dataset

lated motion of the whole body. The dataset contains videos (180*144, 25fps) of the 9 actions in Figure 7 performed by 9 different people.

As in section 4.1, each action is represented by a node in the top-level composite event graph. Each composite event node corresponds to a 2 or 4 node primitive event VTHMM, and each primitive node in turn corresponds to multiple track-level HMMs (one for each body part that moves). Thus the "walk" event has 4 primitive events (one for each half walk cycle) and 6 track HMMs (for the right and left shoulder, upper leg and lower leg), while the "bend" event has 2 primitives (for forward and backward motions) and 3 track HMMs (for the torso, knee and right shoulder) for each primitive. We learn the primitive and track level transition probabilities and also the average velocities for actions involving translation (walk,run,side,jump).

Table 2 compares the performance of our approach with others and figure 8 shows some sample results. As can be seen, while the accuracy and speed are high, the latency is also higher compared to the gesture set. This is because the actions involve more moving parts and hence it takes longer to infer the state at any instant. Also note that even in cases where the initial joint estimates and pose tracking had errors, the recognition layer recovers. Next, we tested the robustness of our method to background, occlusions, style variations(carrying briefcase/bag, moonwalk etc) and also viewpoint(pan angle in the range $0^\circ - 45^\circ$). Our method is fairly robust to all these factors (Figure 9) and table 3 summarizes these results.

[1] reports 99.36% classification rate at ≈ 2 fps on a 3GHz P4 on the same set. Their approach focuses on extracting complex features (3d space-time shapes) from

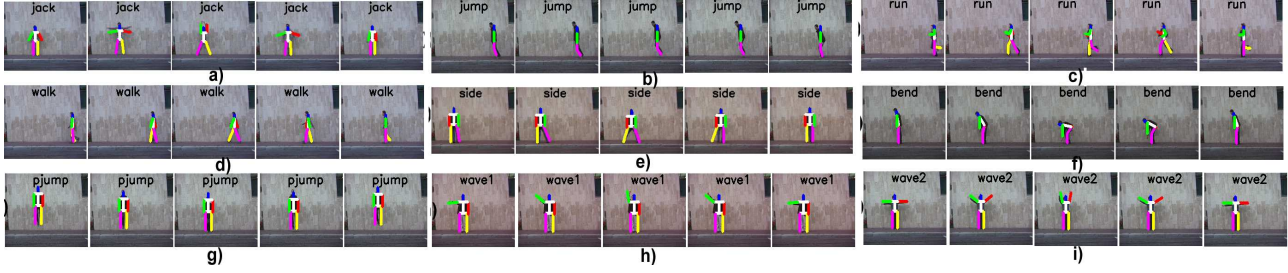


Figure 8. Sample tracking and recognition results - a)jack b)jump c)run d)walk e)gallop sideways f)bend g)jump in place h)wave1 i)wave2

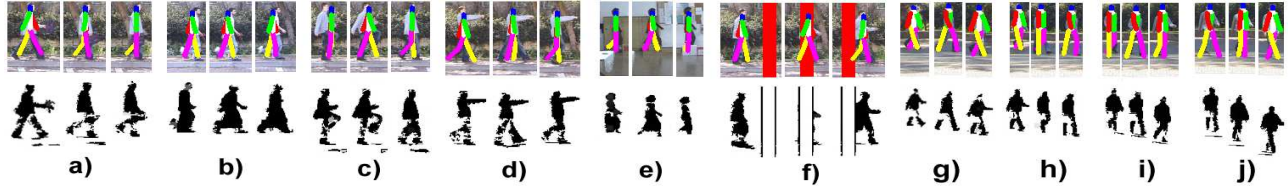


Figure 9. Tracking and recognition with - a)Swinging bag b)dog c)knees up d)moonwalk e)Occluded feet f)Occlusion by pole g)Viewpoint 20° pan h)Viewpoint 30° pan i)Viewpoint 40° pan j)Viewpoint 45° pan

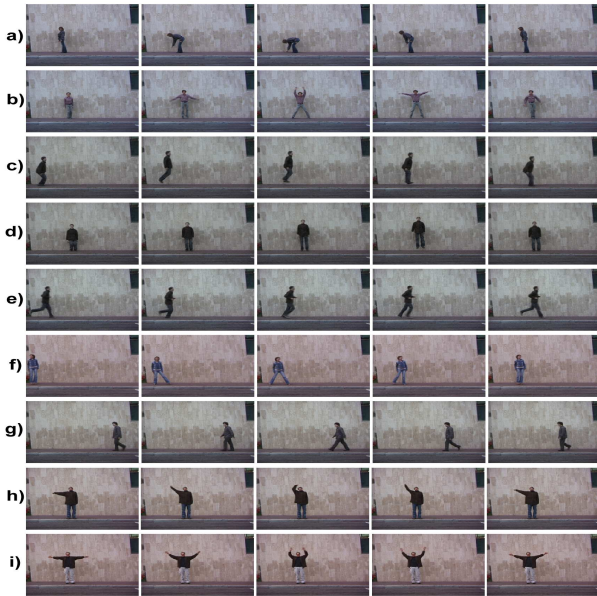


Figure 7. Actions tested in Experiment 2 - a)Bend b)Jack c)Jump d)Jump-In-Place e)Run f)Gallup Sideways g)Walk h)Wave1 i)Wave2

the silhouettes and classifying them using a simple nearest-neighbor approach. Our work on the other hand uses very simple features (foreground silhouettes) and moves the complexity to the higher layers.

Next, to test the generality of our learned models, we ran our method on a few additional walking and jogging sequences on a different dataset(from [12]) *without*

Test Sequence	1st best	2nd best
Carrying briefcase	walk	run
swinging bag	walk	side
Walk with dog	walk	jump
Knees up	walk	run
Limp	jump	walk
Moon walk	walk	run
Occluded feet	walk	side
Full occlusion	walk	run
Walk with skirt	walk	side
Normal walk	walk	jump

Table 3. Robustness under occlusion, style variations and other factors

re-training. This dataset contains several challenging sequences with jittery camera motion as well as variations in tilt and pan angles. While the initial joint estimates and pose tracking had some errors in these sequences, the recognition rate was mostly unaffected under these variations. Figure 10 illustrates some of these results.

5. Discussion and Future Work

We have introduced a novel graphical model for activity recognition that can model both hierarchy and typical durations, and also algorithms for learning and decoding. The decoding algorithm for this model runs at real-time and is online. We use a top-down *tracking-as-recognition* approach for robust recognition in the presence of low level errors. We demonstrate our approach in the domain of continuous gesture recognition as well as for tracking and

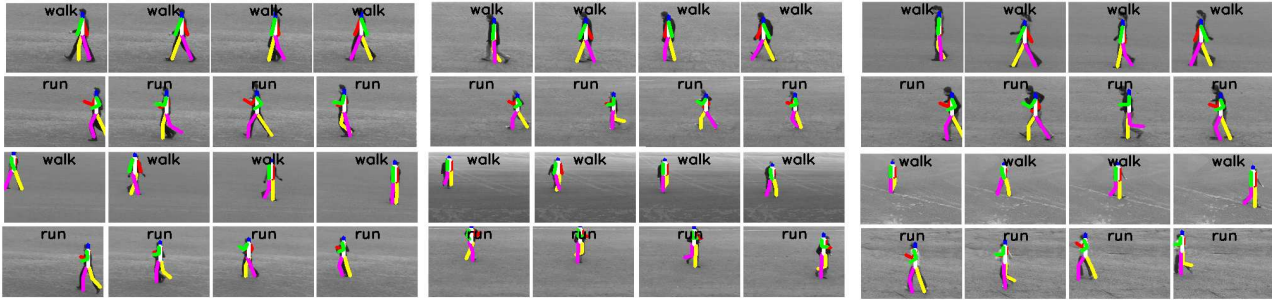


Figure 10. Tracking and recognition with learned models on dataset from [12] with variations in viewpoint as well as jittery camera motion

recognizing articulated human motion. We have shown the robustness of our techniques to variations in background, style, clothing, lighting, occlusions, jittery camera movements and also viewpoint to some extent. Also, although our tracking performance is reasonable it is not perfect and tends to wander a bit. This does not affect our recognition performance as the higher level event models is robust to these errors and also corrects the tracking errors.

While our approach is fairly robust, fast and accurate, it also requires some knowledge of the structure of the actions like the limbs involved. Such event structure could potentially be learned from Mocap data. We also plan to investigate the use of edge-based features instead of the foreground blobs obtained from background subtraction to make our approach robust to large camera motion.

Acknowledgements: We thank the authors of [6, 13] for sharing the gesture dataset. The action datasets were downloaded from the *Berkeley Action Recognition* page². We also thank the authors of [1, 12] for making these datasets public.

References

- [1] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *ICCV*, pages 1395–1402, 2005. 6, 8
- [2] A. F. Bobick and J. W. Davis. The recognition of human movement using temporal templates. *PAMI*, 23(3):257–267, 2001. 2
- [3] M. Brand, N. Oliver, and A. Pentland. Coupled hidden markov models for complex action recognition. *CVPR*, pages 994–999, 1997. 2
- [4] M. Chan, A. Hoogs, R. Bhotika, A. Perera, J. Schmiederer, and G. Doretto. Joint recognition of complex events and track matching. In *CVPR*, pages II: 1615–1622, 2006. 2
- [5] T. Duong, H. Bui, D. Phung, and S. Venkatesh. Activity recognition and abnormality detection with the switching hidden semi-markov model. *CVPR*, 1:838–845, 2005. 2
- [6] A. Elgammal, V. Shet, Y. Yacoob, and L. Davis. Learning dynamics for exemplar-based gesture recognition. In *CVPR*, pages 571–578, 2003. 5, 6, 8
- [7] S. Hongeng and R. Nevatia. Large-scale event detection using semi-hidden markov models. *ICCV*, 2:1455, 2003. 2
- [8] M. Narasimhan, P. A. Viola, and M. Shilman. Online decoding of markov models under latency constraints. In *ICML*, pages 657–664, 2006. 2, 4, 5
- [9] A. Y. Ng and M. I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *NIPS*, pages 841–848, 2001. 2
- [10] N. Nguyen, D. Phung, S. Venkatesh, and H. Bui. Learning and detecting activities from movement trajectories using the hierarchical hidden markov models. In *CVPR*, pages II: 955–960, 2005. 2
- [11] P. Ramesh and J. G. Wilpon. Modeling state durations in hidden markov models for automatic speech recognition. *ICASSP*, pages 381–384, 1992. 2, 3
- [12] C. Schüldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. In *ICPR (3)*, pages 32–36, 2004. 7, 8
- [13] V. Shet, S. N. Prasad, A. Elgammal, Y. Yacoob, and L. Davis. Multi-cue exemplar-based nonparametric model for gesture recognition. In *ICVGIP*, 2004. 5, 6, 8
- [14] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Conditional random fields for contextual human motion recognition. In *ICCV*, pages 1808–1815, 2005. 2
- [15] C. Vogler and D. Metaxas. Parallel hidden markov models for american sign language recognition. *ICCV*, pages 116–122, 1999. 2
- [16] S. B. Wang, A. Quattoni, L.-P. Morency, D. Demirdjian, and T. Darrell. Hidden conditional random fields for gesture recognition. In *CVPR (2)*, pages 1521–1527, 2006. 2
- [17] D. Weinland, R. Ronfard, and E. Boyer. Automatic discovery of action taxonomies from multiple views. In *CVPR*, pages II: 1639–1645, 2006. 2
- [18] T. Zhao and R. Nevatia. 3d tracking of human locomotion: A tracking as recognition approach. *ICPR*, 1:541–556, 2002. 1, 2

²<http://www.eecs.berkeley.edu/Research/Projects/CS/vision/action/>