View and Scale Invariant Action Recognition Using Multiview Shape-Flow Models

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

Abstract

Actions in real world applications typically take place in cluttered environments with large variations in the orientation and scale of the actor. We present an approach to simultaneously track and recognize known actions that is robust to such variations, starting from a person detection in the standing pose. In our approach we first render synthetic poses from multiple viewpoints using Mocap data for known actions and represent them in a Conditional Random Field(CRF) whose observation potentials are computed using shape similarity and the transition potentials are computed using optical flow. We enhance these basic potentials with terms to represent spatial and temporal constraints and call our enhanced model the Shape,Flow,Duration-Conditional Random Field(SFD-CRF). We find the best sequence of actions using Viterbi search in the SFD-CRF. We demonstrate our approach on videos from multiple viewpoints and in the presence of background clutter.

1. Introduction

Recognition of human actions from video sequences is important for a number of tasks including video monitoring, video indexing and human-compuer interaction. There has been significant amount of research in activity recogntion in recent years; however, the ability of the current systems remains limited. One of the key limitations comes from the assumption of accurate low-level tracking, as in ([4]) or extraction of clean silhouettes as in ([15][11][17]); this is not always possible under realistic operating conditions. There are also methods that avoid the step of body or limb tracking, such as ([14, 10]); however, these methods are typically not invariant to viewpoint and scale variations.

We present a method that combines the steps of tracking and event recognition. We use a graphical representation to represent multiple events, with each event being described by a pose sequence. To accomodate variations in appearance due to viewpoint, the model includes appearances at different viewing angles. Our observations include multiple low-level features such as a pedestrian detector, matches of image edges with model silhouettes and motion flow features. This results in a system that is robust to variations in viewpoint, imaging conditions and the background environment.

Our focus has been on single human activities such as $sitdown$, $standup$, $pickup$, $point$, etc. where the human's location is relatively fixed. We believe that the method is not limited to the actions described and can also be generalized to actions involving locomotion and multiple people.

1.1. Related Work

Approaches to activity recognition can be classified as being in two broad threads - the first starts with image level features (like shape or optical flow) and recognizes events by comparing the image features to a set of event templates, while the second focuses on modeling the high-level structure of events with graphical models

Template based approaches focus on extracting low-level image features which are then compared to features that are pre-extracted from a set of event templates for recognition. [1] introduced *motion energy images* for correlating view-based action templates with foreground images. [4] describes recognition of actions at a distance by correlating optical flow templates with track windows of a stabilized human figure. [14] presents an approach to compare two space-time intensity patterns without explicitly computing the optical flow. In contrast to these flow based templates, [5] used shape based templates, and applied them for recognizing arm gestures. In recent work, [10] uses a combination of shape and flow features to show encouraging results for event detection in several cluttered scenes. While these approaches offer a direct way to model high-level events in terms of image features, they are also highly viewpoint and

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

scale dependent.

Graphical models on the other hand provide a natural framework to represent state transitions in events and also the spatio-temporal constraints between the actors and events. Hidden Markov models (HMM) and their extensions have been widely used in various domains successfully. [3] introduced the switching hidden semi-Markov model (S-HSMM) to simultaneously model both the natural hierarchical structure as well as durations of events. [11] introduced *ActionNets* that uses keyposes of actions rendered from multiple viewpoints for view-invariant action recogntion. *Discriminative* models like conditional random fields (CRF) are becoming increasingly popular due to their flexibility and improved performance. [15] applied CRFs for contextual motion recognition and showed encouraging results. [12] introduced a 2-layer extension (LDCRF) to the basic CRF framework and applied it for continuous gesture recognition. While each of these models provide a framework for modeling different aspects of actions, there is a large gap between most activity models and image data. This gap is typically bridged by using fairly accurate tracks from an intermediate module ([2, 3, 12]) or by extracting features from clean silhouettes ([15, 11]).

1.2. Overview of our Approach

In our work, we combine ideas from the graphical model and template based threads and demonstrate our approach on videos with large variations in viewpoints and scale and also in the presence of background clutter. Similar to [11] we first render Mocap data of various events in multiple viewpoints using *Poser*. We then embed these templates into a 2 layer graph model similar to [12, 3]. The nodes in the top layer correspond to events in each viewpoint and the lower layer corresponds to each pose in the event. At each frame we compute the observation probability based on shape similarity using the scaled-Hausdroff distance and the transition probability based on flow similarity using features similar to [4]. We augment the similarity score with a duration term to account for events taking place at different speeds, and a spatial term that provides a *Kalman filter* like framework for tracking the person. We recognize events using Viterbi search on the graphical model. Our approach for simultaneously tracking and recognizing actions also builds on the *tracking-as-recognition* approach in [19].

The rest of the paper is organized as follows - First we give an overview of our setup to generate the multiview templates and the high level constraints for representing actions, in section 3 we describe our graphical model for inferring the event and pose sequence and in sections 4 we describe our shape and flow similarity measures respectively. Finally we present results of our system in section 5 and conclude in section 6.

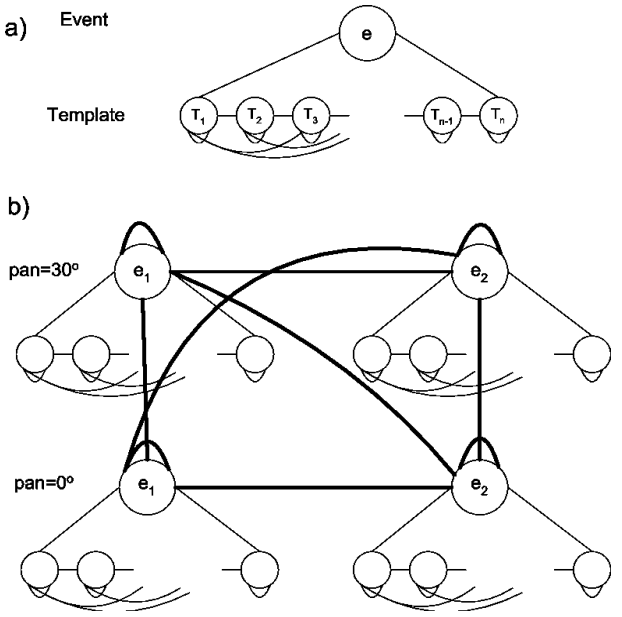

Figure 1. Transition Constraints - a) Graph model for a single event b) 2-layer model for a simple 2-event recognizer at the first two pan angles (0^o,30^o)

2. Action Representation

Human actions involve both spatial (represented by the pose) and temporal (corresponding to the evolution of body pose over time) components in their representation. Further, the actual appearance of the spatio-temporal volume varies significantly with scale and viewpoint. In order to make our representation invariant to viewpoint, we first render poses of synthetic human figures from *motion capture* data (obtained from [6]) of various actions in multiple viewpoints using *POSER* [1]. We cover 90^o of camera tilt angle at 15^o intervals and 360^o of pan at 30^o intervals. We render our poses with a large resolution (900×600 pixels) and use a scale invariant distance measure to make our approach robust to variations in scale. Further, we include body poses in all frames of the action template instead of just the key poses as in [11]. This is because we use flow based measures besides shape, and hence using key poses with a large pose difference would make the flow matching very inaccurate.

We embed the poses in a 2-layer graphical model illustrated in Figure 1. Each node in the top layer corresponds to an action in a particular viewpoint and the lower layer corresponds to the individual poses. We restrict transitions at the event layer based on the similarity between the low level poses at the transition point. For example, we can transition to the *standup* only after a *sitdown* event since the final pose in *sitdown* is very similar to the start pose in

[1] POSER 5, Curious Labs (now e frontier Inc.)

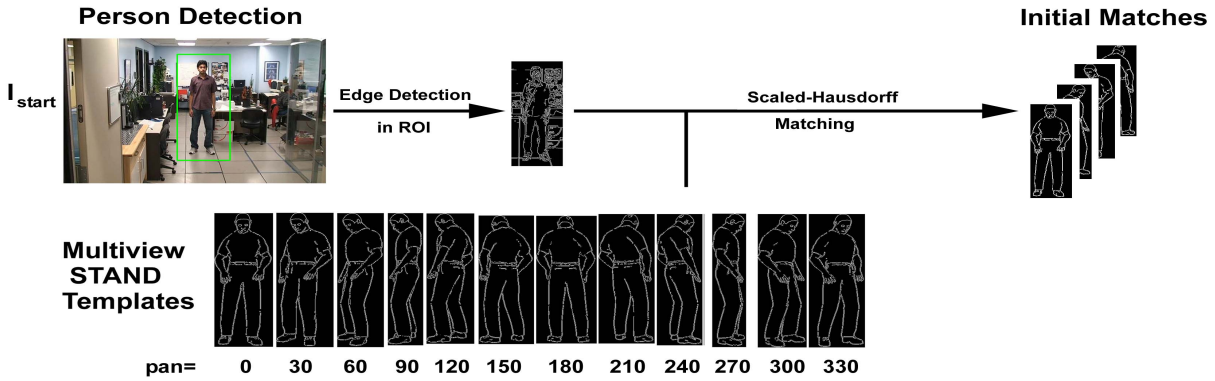

Figure 2. Pose intialization starting with an initial detection window (Green Box in I_{start})

standup, while we can transition to several events from the *stand* action (with a single $stand$ pose at the lower layer). Further, at the lower layer we restrict pose transitions based on the expected speed of the top layer event. Thus, for the *sitdown* event one cannot transition directly from the starting stand pose to the last sitting pose, but must go through the intermediate poses. To reduce the complexity of inference, we assume that the approximate tilt angle is known and we need to only consider different pan angles. This is reasonable in most applications where the tilt is fixed and only the relative pan of the actor to the camera varies.

We compute the similarity between the image sequence and the event templates by embedding shape and flow similarity scores and also the transition constraints in Figure 1 into the observation and transition potentials of a CRF. Since our model includes shape and flow features and also models event durations, we call it the *Shape,Flow,Duration-CRF(SFD-CRF)*. CRF is a generalization of HMM that allows observation and transition potentials to be arbitrary functions that can vary with position in the sequence. Further, the observation and transition potentials need not have a probabilistic interpretation making it ideally suited for embedding our shape and flow similarity measures. We describe the details of the SFD-CRF in the next section.

3. Pose Tracking and Recognition

During recognition, we start with a human detection window obtained with a state-of-the-art pedestrian detector([18]) and use the edge map within the window as our basic observation. The detector does not precisely segment the human form and also does not give orientation information. So we first refine the detection by matching templates of the standing pose in multiple orientations, within the detection window at different scales (in our experiments, we looked at 3 scales below the height of the detection window at steps of 1.1). Figure 2 illustrates the pose initialization process described. We then track and recognize the events by traversing the graph model in Figure 1 starting from the initial set of poses.

Let $\mathbf{I} = \{I_1,I_2,...,I_T\}$ denote the sequence of frames in the video, $\mathbf{e} = \{e_1,e_2,...,e_T\}$ denote the sequence of $[event, viewpoint]$ tuples through the top layer of the graphical model in Figure (1) (since we input the tilt angle the viewpoint corresponds to the possible pan angles), $\mathbf{p} = \{p_1,p_2,...,p_T\}$ denote the sequence of pose templates through the lower level and let $\mathbf{w} = \{w_1,w_2,...,w_T\}$ denote the sequence of track windows for the actor through the video. The state θ_t of the person at frame t is denoted by the tuple $[e_t,p_t,w_t]$. Then, the probability of the state sequence $\theta = \{\theta_1,\theta_2,...,\theta_T\}$ given the observation sequence $\mathbf{I}$ is given by the standard CRF formulation-

$$P(\theta|\mathbf{I}) = \frac{1}{Z}\phi(\theta_1, I_1)\prod_{t=2}^{T}[\psi(\theta_{t-1}, \theta_t, I_{t-1}, I_t)\phi(\theta_t, I_t)] \quad (1)$$

where, $\phi(\theta_t, I_t)$ is the observation potential, $\psi(\theta_t, \theta_{t-1}, I_t, I_{t-1})$ is the transition potential and Z=$\sum_\theta P(\theta|\mathbf{I})$ is an observation dependent normalization factor. Equation 1 is very similar to the *Conditional Random People(CRP)* formulation presented in [16] for tracking people using a set of pose templates. Our approach differs from [16] in 3 ways - first, we simultaneously recognize and track the actions. Hence the pose transitions depend not only on their similarity but also on the spatio-temporal constraints imposed by the action. Second, since we are interested in recognition our pose tracking is coarse and does not include the *Grid Filtering* done in [16]. Third, our similarity features are scale invariant while they assume that silhouette is scaled and centered $apriori$.

Since we start with a person detection in the first frame, p_1 corresponds to the standing pose. The observation potential $\phi(\theta_t, I_t)$ is measured using the shape similarity measure defined later in equation (14) within the current track window. As the shape similarity measure depends only on the pose template p_t and the track window w_t, we

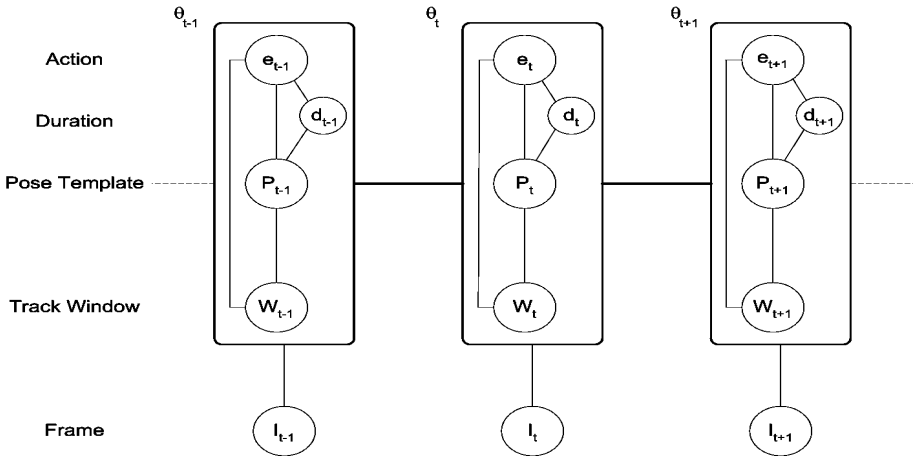

Figure 3. Unrolled Graphical Model of the SFD-CRF for Pose Tracking and Recognition

have-

$$\phi(\theta_t, I_t) = \phi([p_t, w_t], I_t) \tag{2}$$

We define the transition potential $\psi(\theta_t, \theta_{t-1}, I_t, I_{t-1})$ as the product-

$$\psi(p_t, p_{t-1}, I_t, I_{t-1}) = \psi_{trans}([e_{t-1}, p_{t-1}], [e_t, p_t]) \\ \psi_{flow}([p_{t-1}, w_{t-1}], [p_t, w_t], I_{t-1}, I_t) \tag{3}$$

where $\psi_{trans}([e_{t-1}, p_{t-1}], [e_t, p_t])$ corresponds to the transition constraints imposed by the high level graph model similar to the one shown in Figure 1 and $\psi_{flow}([p_{t-1}, w_{t-1}], [p_t, w_t], I_{t-1,I_t})$ is defined using the flow similarity defined later in equation (15) which can be computed given the track windows and the pose templates.

There are two key issues with these basic potentials - first, since there are typically several noisy edges in the image besides the person, one can match on some background edges and stay in a specific pose. Second, in any action the actor tends to move around and hence we must allow for some motion of the track window. However, such moves can accumulate over time and the track window can wander off by matching on the background edges.

We address the first problem by augmenting the state at each frame with a duration node and adding a temporal penalty term to the observation potential in equation (2) that models speed at which an action takes place. Thus the state θ_t corresponds to the tuple [e_t,p_t,w_t, d_t] where d_t is the duration for which the actor has been performing action e_t. We model the speed of action with a Gaussian whose parameters can be learned from the Mocap data. If p_t is the i^{th} pose under event e_t, the temporal penalty $\phi_{time}(e_t, p_t, d_t)$ is given by:

$$\phi_{time}(e_t, p_t, d_t) = \frac{1}{\sigma_t\sqrt{2\pi}} e^{-\frac{(i/d_t - \mu_t)^2}{2\sigma_t^2}} \tag{4}$$

where, μ_t is the mean speed and σ_t is the standard deviation for the action e_t. These can be learned by finding the mean and standard deviation of the lengths of action segments in the Mocap data. For example, if the $sitdown$ action template that we use consists of a sequence of 74 pose templates and the average length of the $sitdown$ action in the Mocap data is ≈60 frames, then the μ_t for $sitdown$ is 74/60=1.23. $\phi_{time}(e_t, p_t, d_t)$ effectively limits the possible poses p_t for any given [e_t,d_t] preventing the action from getting stuck at any pose. This term also plays a role similar to the "Blurry I" kernel used in [4] to allow actions to occur at different rates.

We address the second problem by scanning a region around the previous template position and choosing the location with the best shape similarity score. In order to prevent the track window from wandering off, we augment the shape similarity score with a multi-variate Gaussian based on the distances moved in the x and y directions by the track window w_t:

$$\phi_{space}(e_t, w_t) = \frac{1}{2\pi\sigma_x\sigma_y} e^{-\Delta x^2/\sigma_x^2} e^{-\Delta y^2/\sigma_y^2} \tag{5}$$

In our experiments, we set σ_x and σ_y to be 20% of the scaled template width and height respectively. The use of Gaussians to define $\phi_{space}(e_t, w_t)$ effectively provides a Kalman filter like mechanism for choosing the track window w_t. While we consider only actions occuring in place in this paper, this potential can be extended to allow actions involving large translations of the actor by estimating the difference between the expected position of the window and the actual position.

Combining equations (2), (4) and (5) we can define the augmented observation potential $\hat{\phi}(\theta_t, I_t)$ as-

$$\hat{\phi}(\theta_t, I_t) = \phi([p_t, w_t], I_t)\phi_{time}(e_t, p_t, d_t)\phi_{space}(e_t, w_t) \tag{6}$$

Figure 3 illustrates the unrolled graphical structure of our model with the temporal and spatial nodes. Each maximal clique in figure 3 corresponds to a potential defined in equations (2)-(6).

With these potentials, the best state sequence can be inferred by computing the maximum probability path-

$$p* = arg\max_{\theta} P(\theta|\mathbf{I}) \tag{7}$$

Equation (7) can be solved using Viterbi-like search. However, since we render $\approx$ 13000 poses and the track window can potentially be at any location in the image (captured at 740×480 pixels resolution) the state space is huge making the computation time impractical. But, given the transition and spatial constraints imposed by the graphical model, only a very small number of these states have significant probability. In our experiments we considered only the top $P = 10$ [p_t,w_t] tuples for each e_t. Since we use templates for 6 actions rendered in 12 possible pan angles, we need to consider only 6*12*10=720 states at each frame to find the best state sequence. Also, since we are only interested in finding the best path through the SFD-CRF, we can ignore the normalization factor Z(I) in equation 1 since it is constant for a given video segment.

4. Shape and Flow Potentials

We now describe the two key potentials in the SFD-CRF.
Shape Matching: The *Hausdorff* measure[7] has been popular for matching images due to its simplicity and extensions that are invariant to translation, scale and rotation. For two sets of points A and B, the directed Hausdorff distance from A to B is:

$$h(A,B) = \max_{a\in A}\min_{b\in B} \parallel a - b \parallel \tag{8}$$

where $\parallel \ . \ \parallel$ is any norm (L1 in our case). In order make the distance robust to outliers, we typically take the partial distance as in [7]-

$$h(A,B) = K^{th}\max_{a\in A}\min_{b\in B} \parallel a - b \parallel \tag{9}$$

where $K^{th}\max$ refers to the K^{th} largest value of $\parallel a - b \parallel$. Typically, we wish to match a set of template edge points B to edges in image A at various image locations and independent x and y scales. Let the quadruple $t = (t_x, t_y, s_x, s_y)$ denote a model transformation $t(B)$. Then [8] presents a scaled Hausdorff distance at t as-

$$h(A,t(B)) = K^{th}\max_{a\in A}\min_{b\in B} \parallel a - (s_x b_x + t_x, s_y b_y + t_y) \parallel \tag{10}$$

While, the Hausdroff score can be used to measure the similarity of particular set of image points to the model, it does not directly give a probability for the match between A and B. In previous work, two probabilisic formulations have been used. The first is the *Hausdorff fraction* which counts the fraction of model points that are at a distance less than a threshold:

$$h_K(A,B) \leq \delta \tag{11}$$

While this is a straight forward generalization of the partial Hausdorff distance in equation (9), it is quite sensitive to the threshold δ. [13] presents an alternative formulation based on the distance of each model point to the nearest image point as follows-

$$P(A|t(B)) = \prod_{i=1}^{|B|} p(D_i) \tag{12}$$

where $p(D_i)$ is a probability distribution function for the distance of each model point to the nearest image point and is defined as-

$$p(D_i) = c_i + \frac{1}{\sigma\sqrt{2\pi}} e^{-D_i^2/2\sigma^2} \tag{13}$$

This formulation was used in [5] to match shape templates using the chamfer distance, for gesture recognition. In our case, since we render the pose templates at a high resolution, the model typically has $\approx$ 1000 points. Hence the RHS in equation (4) tends to zero even for well matched templates. Instead we first compute the scaled Hausdorff distance for the entire template using equation (10) and then embed it in a normal distribution, to define our shape similarity potential $\phi([p_t, w_t], I_t)$:

$$\phi([p_t, w_t], I_t) = P(A|t(B)) = \frac{1}{\sigma\sqrt{2\pi}} e^{-h(A,t(B))^2/2\sigma^2} \tag{14}$$

In our experiments we set $\sigma = 15$ though the results are fairly robust to the actual value of σ.

Flow Similarity: We measure flow similarity between the event templates and the video based on pixel-wise optical flow, similar to [4]. However, our approach differs from [4] in two crucial aspects - 1) [4] assumes that the actor in the action is already tracked and stablized, while we explore a set of possible windows at each time step and thus simultaneously do tracking and recognition 2) There is a large difference in scale between the templates and the actor in the image. Hence we cannot precompute the template flows. We will discuss how we address the first problem in the next section. Here we will focus on computing a flow similarity given two templates T_1, T_2 and two image windows in consective frames I_{t-1}, I_t at frames t and $t-1$.

In order to compute the optical flow between two templates, we first scale them based on the image windows and

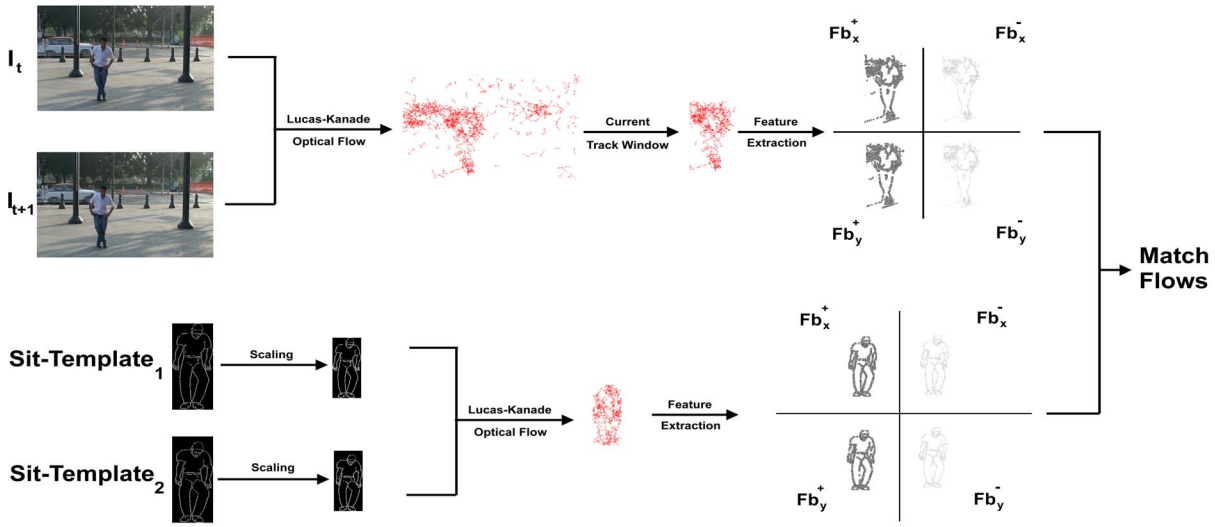

Figure 4. Matching optical flow in image with template optical flow

then compute optical flow using the Lucas-Kanade [9] algorithm. Then, similar to [4], we split the optical flow vector field $\mathbf{F}$ into two scalar fields F_x and F_y corresponding the x and y components, then half-wave rectify them into four non-negative channels F_x^+, F_x^-, F_y^+, F_y^- and finally blur and normalize them with a Gaussian to obtain the final set of features $\hat{Fb_x^+}$, $\hat{Fb_x^-}$, $\hat{Fb_y^+}$, $\hat{Fb_y^-}$. We extract a similar set of features from the image windows and compute the flow similarity as-

$$\psi_{flow}(T_1, T_2, I_{t-1}, I_t) = \frac{1}{4}\sum_{c=1}^{4} \frac{\sum_{x,y \in I} a_c(x,y) b_c(x,y)}{|a_c||b_c|} \quad (15)$$

where I refers to the spatial extent of the flow descriptor, the b_c's refer to the features extracted from the templates and the a_c's refer to the image features. Note that equation (15) is normalized to be in the range [0,1]. Figure 4 illustrates the computation of the flow similarity described.

5. Experiments

We tested our approach on videos of 6 actions - *sit-on-ground(SG)*, *standup-from-ground(StG)*, *sit-on-chair(SC)*, *standup-from-chair(StC)*, *pickup(PK)*, *point(P)*. We collected instances of these actions around 4 different tilt angles - 0^o, 15^o, 30^o, 45^o. We did not precisely calibrate the camera at each tilt and typically had a tilt error of $\approx 5^o$. At each tilt we collected instances of actions at 4 different pan angles - typically, around 0^o, 45^o, 90^o, 270^o, 315^o though the actual pan was not measured precisely. We collected one instance of each action for each [tilt,pan] combination from four different actors for a total of 16 instances of each action at each tilt. Further for tilt=0^o, we collected videos under 6 widely varying backgrounds including indoors in office environments and outdoors in front of moving vehicles for a total of 24 instances of each action at that tilt. In all we had 400 instances of all actions across all tilts and pans. We also varied the zoom of the camera and hence the actual size of the person varied between ≈80-300 pixels in 740×480 resolution videos. The pose templates were rendered so that the standing pose is ≈ 600 pixels tall. Figure 5 illustrates some of the conditions under which we tested our approach.

To process the videos, we first apply a pedestrian detector similar to [18]. As the detector is trained only for the standing pose, it fails when the pose of the actor changes during an action. Thus the detections provide an approximate segmentation of the event boundaries in the video sequence. We tested our algorithm by running our recognizer between two detections and then comparing the highest probability event sequence in the intervening frames to the ground truth.

To measure the relative importance of $shape$ features, $flow$ features and $duration$ modeling we compare our system ($shape + flow + duration$) with using only $shape$, only $flow$ and $shape + flow$ potentials in the CRF. Table 1 summarizes the accuracy at different tilt angles. Note that using only $flow$ features is very similar to [4] except that we don't start with track windows that are centered and scaled. All the speed numbers reported are for the entire system including detection, tracking and recognition and were obtained by running C++ Windows programs on a single 2GHz Pentium IV CPU.

As can be seen, combining $shape$ with $flow$ features produces a significant improvement over using either of these alone. The result is further improved by modeling event durations. Also note that including duration models

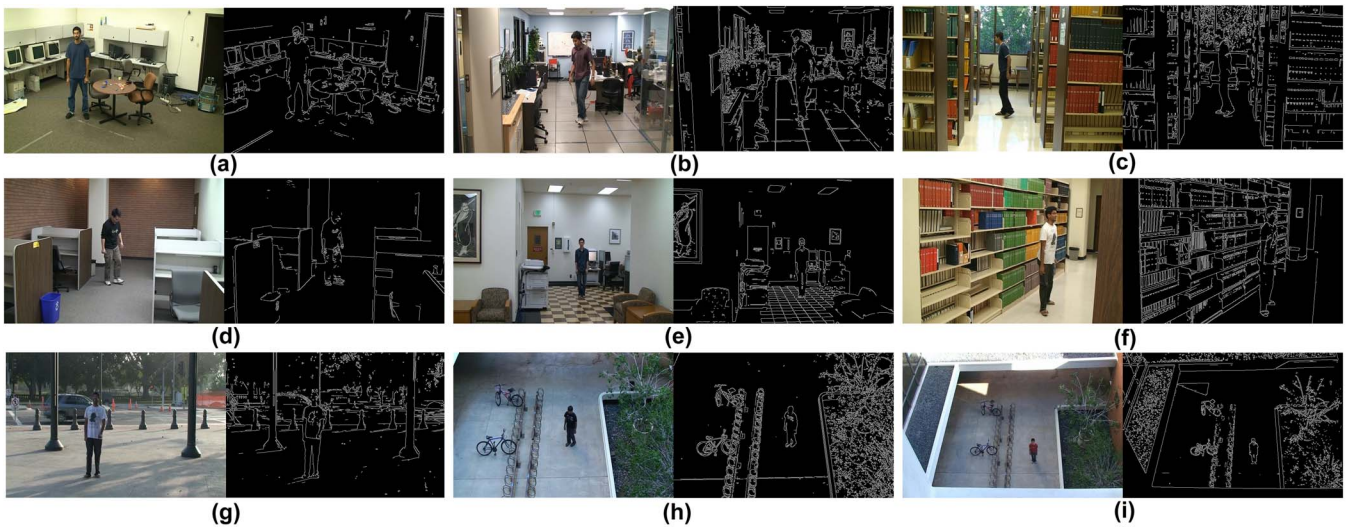

Figure 5. Sample Background, Viewpoint and Scale variations tested - (a)Indoor office environment,tilt=15^o,pan=0^o (b)Indoor office environment,tilt=0^o,pan=30^o (c)Indoor library,tilt=0^o,pan=90^o (d)Indoor office,tilt=0^o,pan=315^o (e)Indoor office,tilt=0^o,pan=0^o (f)Indoor library,tilt=0^o,pan=270^o (g)Outdoor with moving cars,tilt=0^o,pan=0^o (h)Outdoor,tilt=30^o,pan=30^o (i)Outdoor,small scale,tilt=45^o,pan=0^o

	0^o	15^o	30^o	45^o	Overall	Speed(fps)
$shape + flow + duration$	77.35	82.98	81.25	65.63	78.86	0.37
$shape + flow$	70.68	76.67	77.42	62.5	72.37	0.34
$flow$	63.79	56.67	61.29	53.12	59.12	0.41
$shape$	56.82	59.01	75.76	56.25	61.18	1.7

Table 1. Comparison of accuracy and speed with $shape$, $flow$, $shape + flow$, $shape + flow + duration$ features at different tilt angles

improves the speed too since they restrict the set of possible poses to consider. However,the cost of computing flow features is high. Since the scale of the actor is unknown, these cannot be pre-computed $apriori$ unlike in approaches which assume a known fixed scale. We alleviate this cost partly by storing the $N = 1500$ most recent template flows computed. At each frame, we first check if a particular template flow is already in the stored set and compute the flow only if it is not present.

Another observation from Table 1 is that the accuracy can vary with tilt angles. At higher tilts the *sit-on-ground* and *pickup* actions look very similar causing a large confusion. Variation in pan angles at a given tilt however does not significantly affect the performance since the actions have very distinct signatures at different pan angles. Table 2 shows the overall confusion matrix across all viewpoints.

	SG	SC	PK	P	StG	StC
SG	75.56	4.44	20.0	0.0	0.0	0.0
SC	11.54	76.92	11.54	0.0	0.0	0.0
PK	15.9	0.0	86.1	0.0	0.0	0.0
P	2.56	0.0	10.26	87.18	0.0	0.0
StG	0.0	0.0	20.0	0.0	75.56	4.44
StC	0.0	0.0	11.54	0.0	11.54	76.92

Table 2. Overall Confusion Matrix

6. Summary and Future Work

We have presented a robust approach for simultaneous tracking and event recognition that embeds low-level shape and optical flow features into a high-level graphical model representation of the actions. We have presented good results under several challenging variations in background, scale and viewpoint. We also show that combining the low-level features produces a significant performance improvement over using either of them alone. Duration models further improve the accuracy and marginally reduce computation time.

The spatial potential $\phi_{space}(e_t, w_t)$ can be extended to handle actions involving large translations of the actor like ($walk$, run etc) by using a velocity model. Such an extension would consider the difference between the current track window position and the expected position to chose the next set of windows. While we have focused on single person actions occuring in place, our method can be extended to

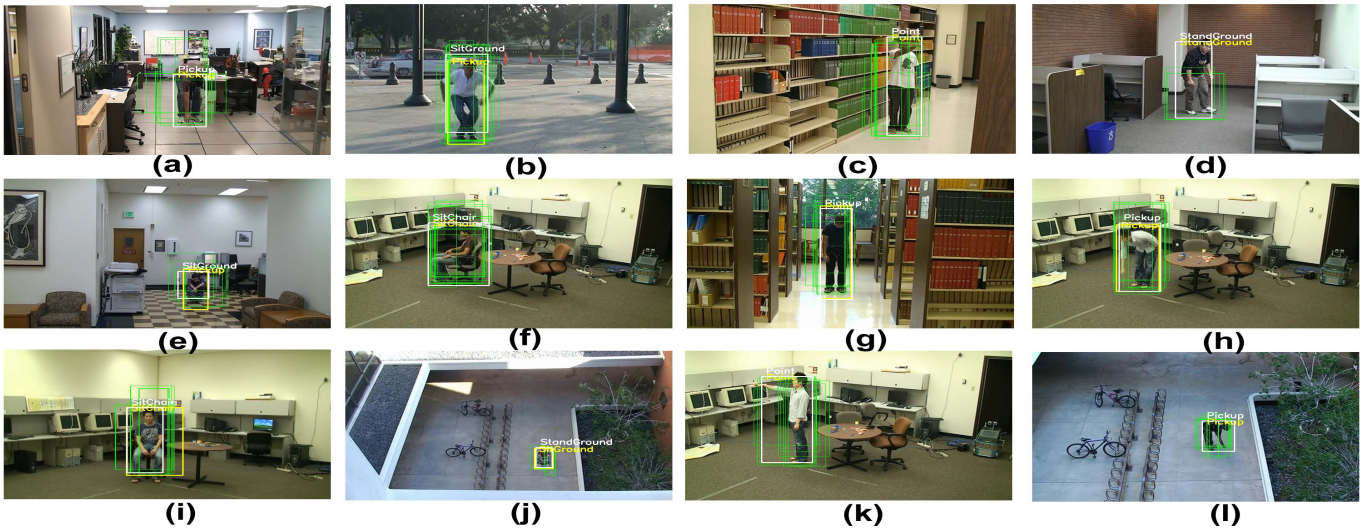

Figure 6. Sample recognition results - green boxes correspond to tracks under consideration, yellow is the best event,pose at that instant and white corresponds to the best event,pose after Viterbi search

a multi-person scenario by detecting each person and then running our recognizer on each detection window. Such an extension would also require explicit occlusion analysis in case of interacting people.

7. Acknowledgment

The authors would like to thank Mr.Jun Yamadera for kindly providing the *Mocap* data used. This research was supported, in part, by the Office of Naval Research under Contract #N00014-06-1-0470 and in part, by the U.S. Government VACE program.

References

[1] A. F. Bobick and J. W. Davis. The recognition of human movement using temporal templates. *PAMI*, 23(3):257–267, 2001.

[2] M. Brand, N. Oliver, and A. Pentland. Coupled hidden markov models for complex action recognition. *CVPR*, pages 994–999, 1997.

[3] 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.

[4] A. A. Efros, A. C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *ICCV*, pages 726–733, 2003.

[5] A. Elgammal, V. Shet, Y. Yacoob, and L. Davis. Learning dynamics for exemplar-based gesture recognition. In *CVPR*, pages 571–578, 2003.

[6] http://www.mocapdata.com.

[7] D. P. Huttenlocher, G. A. Klanderman, and W. Rucklidge. Comparing images using the hausdorff distance. *PAMI*, 15(9):850–863, 1993.

[8] D. P. Huttenlocher and W. Rucklidge. Multi-resolution technique for comparing images using the hausdorff distance. In *CVPR*, pages 705–706, 1993.

[9] T. Kanade and B. Lucas. An iterative image registration technique with an application to stereo vision. In *IJCAI*, pages 674–679, 1981.

[10] Y. Ke, R. Sukthankar, and M. Hebert. Event detection in crowded videos. In *ICCV*, 2007.

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

[12] L.-P. Morency, A. Quattoni, and T. Darrell. Latent-dynamic discriminative models for continuous gesture recognition. In *CVPR*, 2007.

[13] C. F. Olson. A probabilistic formulation for hausdorff matching. In *CVPR*, pages 150–156, 1998.

[14] E. Shechtman and M. Irani. Space-time behavior based correlation. In *CVPR (1)*, pages 405–412, 2005.

[15] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Conditional random fields for contextual human motion recognition. In *ICCV*, pages 1808–1815, 2005.

[16] L. Taycher, D. Demirdjian, T. Darrell, and G. Shakhnarovich. Conditional random people: Tracking humans with crfs and grid filters. In *CVPR (1)*, pages 222–229, 2006.

[17] D. Weinland, E. Boyer, and R. Ronfard. Action recognition from arbitrary views using 3d exemplars. In *ICCV*, 2007.

[18] B. Wu and R. Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *ICCV*, pages 90–97, 2005.

[19] T. Zhao and R. Nevatia. 3d tracking of human locomotion: A tracking as recognition approach. *ICPR*, 1:541–556, 2002.