Accurate Person Tracking Through Changing Poses for Multi-view Action Recognition

Pradeep Natarajan[*]
Raytheon BBN Technologies
10 Moulton St
Cambridge MA 02138, USA
pradeepn@bbn.com

Prithviraj Banerjee
University of Southern California
Los Angeles CA 90089, USA
pbanerje@usc.edu

Ram Nevatia
University of Southern California
Los Angeles CA 90089, USA
nevatia@usc.edu

ABSTRACT

Actions by humans in real-world settings involve large changes in the person's pose and the relative orientation with respect to the camera. Person tracking algorithms often fail under such conditions, since they work by detecting and tracking people in a few known poses (typically standing). Further, due to occlusions and similarity of clothing with background, foreground silhouettes are typically very noisy. We present an approach which address these problems by first accurately tracking a person through changing pose and broken foreground blobs. During the tracking process we also estimate the relative orientation and scale of the person. We represent the pose of the person in each track window using a grid-of-centroids model, and recognize the action by matching with a set of keyposes, in each frame. We tested our approach in a dataset collected in a real grocery store, and report better than ≈82.5% accuracy for frame-by-frame recognition of actions.

1. INTRODUCTION

Systems for automatic analysis and recognition of actions from videos have several compelling applications including human-computer interaction, visual surveillance, search and retrieval among others. While the general problem of action recognition in unconstrained videos is extremely difficult, simple methods that work in well-constrained environments can have a significant practical impact. With this in mind, we focus on the problem of tracking and recognizing single person actions, in largely static background, indoor environments. Such conditions are common in libraries and small grocery stores where most aisles are sparsely populated.

Extensive research has been conducted in action recognition, due to the range of potential applications. However, the applicability of current methods remain limited, since low-level tracking and pose estimation remains extremely hard. While significant progress has been made in pedestrian detection[5][28] and tracking [15], such *detect-and-associate* methods can handle only a few known poses (typically upright). This is because the detectors need a large set of training images for each pose of interest, and hence are not scalable for actions involving extensive pose articulation. This makes it necessary to develop suitable techniques to predict and interpolate between two detections of known poses, which must also be robust to false alarms and missed detections.

Further, many action recognition approaches rely on extracting 2D features like spatio-temporal interest points[13] and learning suitable classifiers. The learned models are typically not generalizable across datasets, viewpoints and scale. This would either require collecting a prohibitively large training set, or retraining the models for each new dataset. To address viewpoint variations, [17][11][20] use 3D *Motion Capture (Mocap)* data of actions rendered from multiple viewpoints, which are then matched to 2D features. However, such *Mocap* data is time consuming to collect and requires expensive equipment. Also rendering the 3D models from multiple viewpoints takes a long time (3-4 days), and requires proprietary software like *Poser*.

We address these limitations by first presenting an algorithm to track a single person through large changes in orientation, scale and pose articulation. During tracking we also estimate the approximate orientation and height of the person with respect to the camera. We represent the pose in each track window of each frame, using a *grid-of-centroids* feature vector that is robust to large errors in background subtraction. We next recognize actions by matching the feature vector with the action models, which is represented using a sequence of keyposes and motion models. The keyposes are represented using stick figures, whose articulations can be specified manually. Further, we learn our motion models for multi-view action recognition from a few video samples, without requiring camera calibration. Our approach presents an effective and easy-to-use framework for frame-by-frame action recognition in common indoor settings. We demonstrate our method on a dataset collected in a real grocery store with 8 different actors and 3 actions, under varying clothing, lighting and viewpoints. Our system achieves ≈82.5% accuracy and runs at 5fps on a standard desktop computer.

In the rest of the paper, we will discuss related work in section 2, overview of our approach in section 3, tracking algorithm in section 4, methods for learning keyposes and motion models in section 5, matching and recognition in section 6 and results in section 7.

[*] Corresponding author

ICVGIP '10, December 12-15, 2010, Chennai, India
Copyright 2010 ACM 978-1-4503-0060-5/10/12 ...$10.00.

2. RELATED WORK

While a wide range of approaches to action recognition have been considered, they can be divided into two broad classes - *Template based* approaches, which focus on extracting low-level features and *model-based* approaches which focus on modeling high-level structure and constraints.

Template based approaches focus on extracting low-level image features which are then either used to train suitable classifiers, or are compared to a set of event templates for recognition. *Motion energy images (MEI)* were used in [2] for correlating view-based action templates with foreground images. This was extended to *motion energy volumes (MEV)* in [27] for 3D action recognition in a multi-camera setup. Shape based templates, instead of foreground images, were used in [9][23] for recognizing arm gestures. Optical flow templates were used in [8] for recognizing actions with track windows of a stabilized human figure, while [1] compared two space-time intensity patterns without explicitly computing the optical flow. More recently, [12] used a combination of shape and flow features for event detection in several cluttered scenes. In contrast to these approaches, *bag-of-words* classifiers trained using local features extracted around a sparse set of interest points, have become popular due to their simplicity and good performance. Spatial corner detectors like the Harris detector were used in [6][18], the sparser *spatio-temporal interest point (STIP)* detectors were used in [13][18][14][4] and a combination of both was used in [22].

The template based approaches directly model high-level events in terms of image features, but need large training sets to generalize across backgrounds, scales and viewpoints. In contrast, several model based approaches that focus on high-level spatio-temporal structure, semantic constraints, and efficient multi-view search, have been considered. In particular, hidden Markov models (HMM) and their extensions have been very popular due to their flexibility and simple Bayesian semantics. HMMs were used in [25][26] for recognizing sentences in American Sign Language (ASL) from hand tracks obtained by tracking colored gloves in video. Coupled-HMM (CHMM) was used in [3] for recognizing multiple interacting actions. The switching hidden semi-Markov model (S-HSMM) was used in [7] to simultaneously model both the natural hierarchical structure as well as durations of events. More recently, *discriminative* models like conditional random fields (CRF) have been used due to their flexibility and improved performance. CRFs were applied for contextual motion recognition in [24], while [19] introduced a 2-layer CRF (LDCRF) and applied it for continuous gesture recognition. These approaches focus on modeling different aspects of actions, but also make several assumptions to bridge the gap with image data. [3][7][19] use fairly accurate tracks from an intermediate module while [24][17] extract features from clean silhouettes, which make then unrealistic for most real applications.

[17] mapped multi-view action templates obtained by rendering *Mocap* data of actions to graphical models for view invariant action recognition. Keypose models of actions from multiple viewpoints were embedded into an *ActionNet.* This attempts to combine the advantages of both template based and model based approaches. The approach is still constrained since they require difficult-to-collect *Mocap* data for each new action. In recent work [21], this method was generalized to include intermediate pose models.

Our approach here addresses these limitations of earlier methods, by using simple keypose action models that do not require *Mocap*, and features that do not require accurate silhouette extraction. Further, since we use multi-view search, our method is robust to viewpoint variations, making it suitable for realistic settings. However, since we do not perform occlusion analysis, our method is restricted to scenarios with a single person or multiple non-occluding persons.

3. OVERVIEW OF APPROACH

Our system consists of the following modules-

2. **Blob Tracking:** At each frame, we run a person detector and also obtain the foreground image. Our tracking algorithm consists of the following steps-
   - *Initialize:* We start with a confident person detection and fit the window on intersecting foreground blobs, to initialize our tracker.
   - *Track:* Once we initialize, we continue tracking by resampling in a region around each previous window, and then fitting the new windows on the foreground.
2. **Model Learning:**
   - *KeyPose selection:* We choose three to five 3D key poses for events of interest by inspection.
   - *Motion model learning:* In addition we learn motion models for each action, in the camera coordinate system. This eliminates the need for accurate camera calibration.
3. **Recognition:** Our recognition algorithm consists of the following steps-
   - We match keyposes with foreground blob of each tracked window.
   - We estimate the person height, width and orientation for each window, and normalize the keyposes using them.
   - We recognize the event corresponding to the best keypose at each frame, by searching around the estimated orientation.

We will now describe each of these steps in detail.

4. TRACKING THROUGH ERRORS AND POSE ARTICULATION

The first step in traditional *bottom-up* action recognition systems is to detect and track the person or object of interest. This is extremely hard in cluttered and crowded environments and is one of the fundamental challenges in computer vision. In recent years, several successful object and pedestrian detection methods have been proposed such as [5][28]. These methods work by first training shape based classifiers from a large set of training images for each object. Given a new image, these classifiers are then used to classify each spatial location to identify possible locations where instances of the object are present. Tracking is then done by *associating* detections in consecutive images using appearance and motion models.

A key limitation of this framework is that many human actions involve extensive pose articulations. The shape and appearance of these poses vary widely, and it is infeasible to collect sufficient examples to train for all possible poses. Hence, we need methods to predict and interpolate the poses between detections of known poses. This was addressed in [20] by using high-level action models. The sequence of poses and actions in the frames between two pedestrian detections were simultaneously inferred using a combination of shape and flow based features, followed by a Viterbi search through the action models.

While effective, this method is *top-down* in the sense that high-level action models drive low-level tracking. Hence they work only in domains with a known *closed-set* of possible actions. The best way to address this is to extract suitable features from each frame of the video and track them. We take a first step at this by combining shape based pedestrian detections with foreground blobs. While blob tracking has a long history in computer vision, their applicability in real indoor settings as input to high-level action recognition is still limited. This is because foreground blobs tend to break up due to errors, occlusions and similarity of clothing with background.

Our approach to track a person through pose articulation and large blob detection errors, works by first detecting the person in a known pose, and then tracking the overlapping blobs by perturbing the previous track window. We will now describe this in detail.

4.1 Track Initialization

Getting an accurate initial track window of a person from foreground blobs is challenging, since a person's foreground can be broken up due to similarity of clothing with background. In many cases, large portions of the person's body can be missing due to similarity of clothing with the background. Shape based pedestrian detectors are robust to such issues, but they typically produce a coarse window as illustrated in figure 1. This is not accurate enough for most action recognition applications.

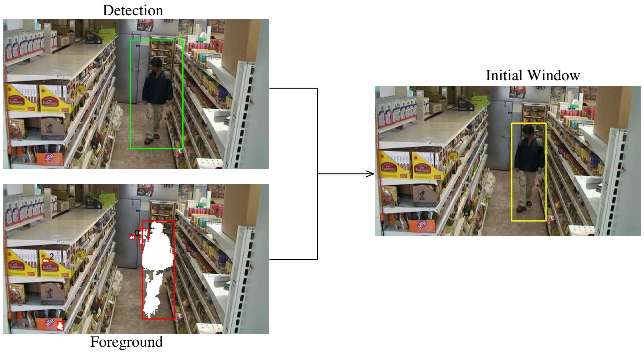

Figure 1: Blob Initialization

To address these issues, we use a combination of shape and blob based detections to initialize our tracker. We start with a *confident* detection from a standard shape-based pedestrian detector, similar to [28]. Next, we extract foreground blobs in the image and fit bounding boxes around each of the extracted contours. We fit the detection window on intersecting foreground blobs, to get a tight fit of the window around the person's body. This is illustrated in figure 1. If there are multiple windows left, we compute an *observation potential* for each window, based on its intersection with the foreground blobs and detections:

$$\phi_{obs} = w_{det}\phi_{det} + w_{fg}\phi_{fg} \tag{1}$$

where, ϕ_{det} and ϕ_{fg} are potentials from the person detection and foreground blobs, and w_{det} and w_{fg} are their relative weights. Also, ϕ_{det} and ϕ_{fg} are defined using the intersections between the final observation window W_{obs} and the detection window W_{det} and foreground windows W_{fg} respectively:

$$\phi_{det} = \frac{|W_{obs} \cap W_{det}|}{|W_{obs}|} \tag{2}$$

$$\phi_{fg} = \frac{|W_{obs} \cap W_{fg}|}{|W_{obs}|} \tag{3}$$

We next store an array of distinct windows W_{obs} sorted according to ϕ_{obs} for tracking the person. In our implementation we set,

$$w_{det} = w_{fg} = 1$$

Note, this framework can be extended to include other features whose relative weights are learned from training data.

4.2 Blob Tracking

Once we obtain the initial window, we track the person through the video using a *CRF-Filter*[16] framework. In the original CRF-Filter framework, we first sample from a transition potential ϕ_{trans} to get possible current states, and then compute the observation potential ϕ_{obs}, followed by re-sampling. Applying this framework directly to person tracking in video is challenging since small errors in the track windows can accumulate causing them to *wander away* from the person. Hence, at each step we re-fit the new windows with the observed foreground blobs. We eliminate identical windows and compute the observation potential. This is illustrated in figure 2.

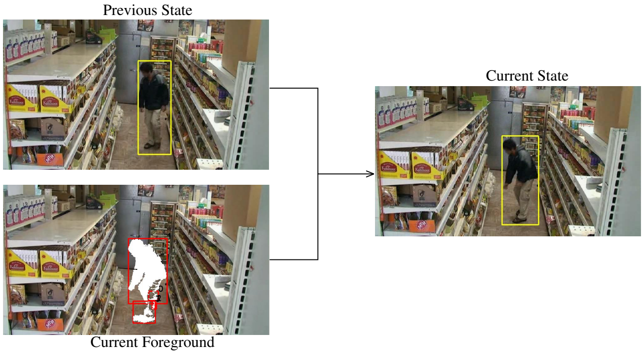

Figure 2: Blob Tracking

Given the previous track window, we obtain the current window by considering different perturbations of it, and fit each one on the current set of foreground blobs, while eliminating duplicate windows. This is described in Algorithm 1. The transition potential $\phi_{trans}(W_t, W^i_{t-1}, O_t)$ corresponds to the *motion model* of the person. In *top-down* approaches,

Algorithm 1 Modified CRF-Filter for Person Tracking

```
1: Inputs: Previous track windows-
   $S_{t-1}=\{\langle W_{t-1}^{(i)},\alpha_{t-1}^{(i)}\rangle | i=1..N\}$
   Observations $O_t=(W_t^{det}, W_t^{fg})$
2: Resampling: Draw $N$ samples $W_{t-1}^{(i)}$ with probability
   proportional to importance weights $\alpha_{t-1}^{i}$
3: for $i=1$ to $N$ do
4:     Prediction: Sample $W_t^i \sim \phi_{trans}(W_t, W_{t-1}^i, O_t)$
5:     Refit: $W_t^i$ with $(W_t^{det}, W_t^{fg})$
6:     Prune: If $W_t^i$ exists in the current set of windows $S_t$
       ignore it; else add to $S_t$.
7:     Importance Sampling: $\alpha_t^i=\phi_{obs}(W_t^i, O_t)$
8: end for
```

we have a set of motion models corresponding to the actions, and we can sample from them to predict the next track window. In our approach we do not know the actions *apriori*. Further, the track windows not only translate but also can change shape due to changing pose. Hence, we perturb *(x,y,w,h)* of each window by sampling uniformly in range [-10%,+10%], and fit them to the observed foreground blobs and detections.

5. MODEL REPRESENTATION AND LEARNING

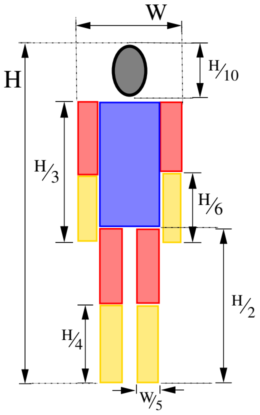

Figure 3: 23D body model

We represent each action by a sequence of 3D *keyposes* using the ideal body model illustrated in figure 3. Our body model has 19 dimensions for the joint angles, with three additional dimensions for direction of translation(x,y,z) and one for scale(H), to give a total of 23 degrees of freedom. Note that we ignore the motion of the head, wrist and ankles as our image resolutions are generally too small to capture them. Each body part is represented as a cylinder, and the pose is fit to the image by projecting it to 2D. The keyposes are selected at key transition points in the action, similar to [17]. This is illustrated in figure 4.

We obtain the keyposes by manually transforming the standing pose to the required keypose. Here, we start with the neutral standing pose and rotate body parts around relevant joints by angles obtained by inspection. The angles are easy to obtain by inspection for many actions like arm gestures, walk run etc.

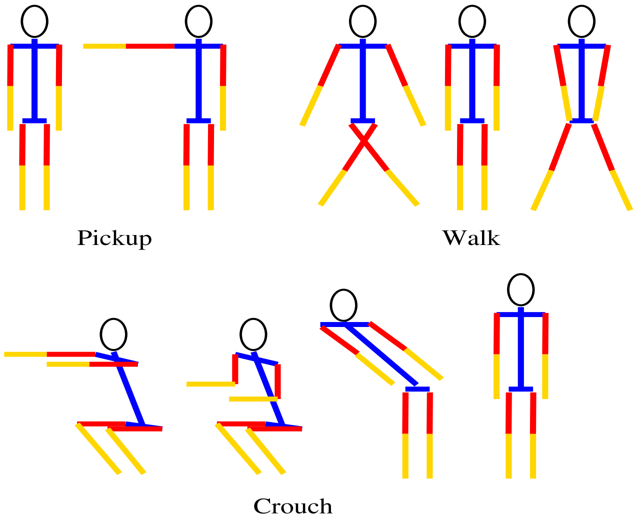

Figure 4: Keyposes for actions recognized

5.1 Motion Model Learning

The speed at which a person moves is an excellent feature for distinguishing between many common actions like stand, walk, run etc. However, it is difficult to estimate the speed if the person moves in a non-fronto-parallel direction, like towards the camera. In such cases, besides translation the person's relative height in the image also changes as illustrated in figure 6. Accurately normalizing for this effect requires precise camera calibration parameters, which is difficult to obtain.

We address these issues by tracking the centroid of the lower 25% of the body, as the rate of displacement of the leg determines a person's speed of motion. This is illustrated in figure 6. To make the system robust to noise and background subtraction errors, we smooth the lower-centroid trajectory over a large window of frames, and then use the frame-by-frame x and y displacements to determine the instantaneous speed. We learn each action's motion model to be a *Gaussian* distribution over the instantaneous speeds of a few (typically 3) actors.

Further, we model the scale change Δh linearly w.r.t the y displacement Δy in the image co-ordinates:

$$h' = h + k\Delta y \tag{4}$$

We learn the constant k from a few sample videos shot at the location of interest. This linear model is a good approximation at low tilt angles, which is valid for most surveillance cameras having ≈15^o tilt. Further, it also allows us to estimate a person's height in image co-ordinates without requiring camera calibration, for highly articulated poses. We use it as an input in our recognition system.

6. RECOGNITION

We track the person through changing poses and do frame-by-frame action recognition. We first estimate the person's height, width and orientation, and infer the action from matching keyposes and motion model.

6.1 KeyPose Matching

Given a track window, we first compute centroids of foreground blobs in a 3*4 (x,y) grid, and ignore grid locations with too few foreground points. Next we compute similar centroids for each key pose, and then compute key pose to track window distance using *scaled Hausdorff* distance [10] between the centroids as follows:

- Let A be the track window centroids and B be be pose centroids.

- Let t=(t_x,t_y,s_x,s_y) denote a translation and scaling of key pose B_e corresponding to event e.

Then the scaled-Hausdorff distance $d_{shape}(A, B)$ is given by

$$\begin{aligned} d_{shape}(A, B_e) = \\ \max\{\max_{a \in A} \min_{b \in B_e} |a - (s_x b_x + t_x, s_y b_y + t_y)|, \\ \max_{b \in B_e} \min_{a \in A} |(s_x b_x + t_x, s_y b_y + t_y) - a|\} \end{aligned} \quad (5)$$

This is illustrated in figure 5.

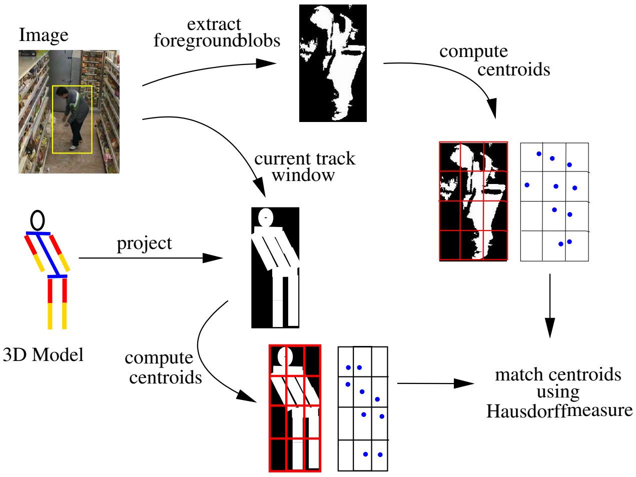

Figure 5: KeyPose Matching using Scaled Hausdorff Distance

6.2 Motion Matching

In addition to the shape matching, we also match the motion model of different actions. At each step in the tracking process, we estimate the person's relative image height, width and orientation to assist our recognition. The method is illustrated in figure 6 and is described as follows :

- If projected person height at location $P_1(x, y)$ is h, height at location $P_2(x{+}\Delta x, y{+}\Delta y)$ is- $h'{=}h{+}k\Delta y$, where k is learnt during motion model learning describe in section 5.1.

- The orientation (relative pan) is approximated to be $tan^{-1}(\Delta y/\Delta x)$.

- We also normalize for different widths among different people. To do this we estimate the *width fraction*, i.e the ratio of width to height, during the first 15 frames of tracking the person and maintain it for the rest of the sequence.

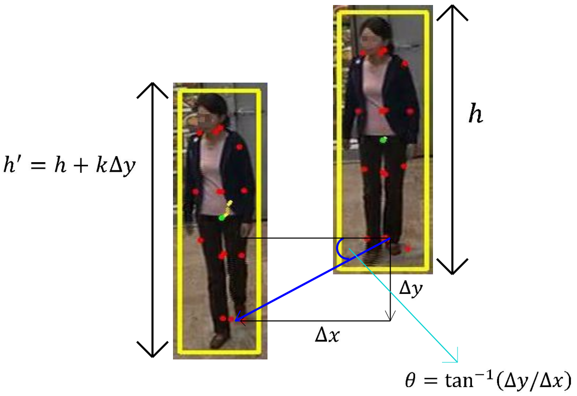

Figure 6: Estimation of relative height and orientation

We compare the distance moved by the actor at each frame with each event's motion model as follows:

$$t = \sqrt{\Delta x^2 + \Delta y^2}$$
$$d_{motion}(e) = -\frac{(t - \mu_e)^2}{2\sigma_e^2} \quad (6)$$

where, Δx and Δy are the displacements in the x and y directions as illustrated in figure 6. μ_e and σ_e are the mean and standard deviations of the motion model for event e, and $d_{motion}(e)$ gives the log-likelihood score.

Given these scores, we recognize the best (*event*,*keypose*) as:

$$\max_{B_e}(w_{shape} d_{shape}(A, B_e) + w_{motion} d_{motion}(e)) \quad (7)$$

where, w_{shape} and w_{motion} are the weights for the shape and motion scores. In our implementation we used,

$$w_{shape} = w_{motion} = 1$$

7. RESULTS

We tested our approach on the Groccery store dataset [21] which is collected in the cluttered setting of a grocery store. The camera is static and has a downward tilt of ≈ 20°. In each video, an actor enters the scene, picks up an item and leaves. The action set includes 3 full body actions - *walking, pickup from shelf (Pickup1), crouch and pickup (Pickup2)*, from 8 different actors for a total of 18 videos and 3657 frames. The observed size of the actor varies from 200 to 375 pixels in a 852 × 480 frame. The main challenges here are poor foreground extraction, highly articulated and ambiguous poses and large changes in the orientation and scale of the actor. The poor foreground extraction is due to the shadows, changes in lighting due to reflection from outside traffic and color similarities between actor's clothing and the background. Sample results are shown in figure 7, which demonstrates our method's effectiveness in tracking humans

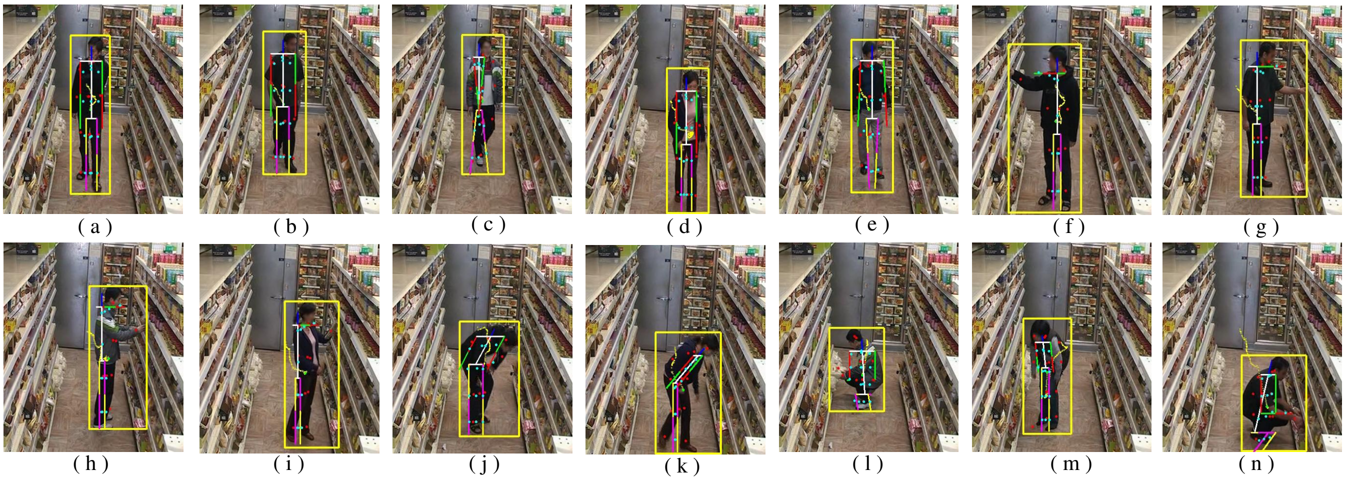

Figure 7: Sample results from Grocery dataset showing Walk (a-e), Pickup1 (f-i) and Pickup2 (j-n) actions.

through complex pose variation, and also inferring the action.

Table 1 summarizes the accuracy on the grocery set, with the numbers in the brackets indicating the actual number of frames in each entry. The entire system runs at ≈$5fps$ on on a 3 GHz Xeon CPU running Windows/C++ programs and include all the steps. Most of the computation arises from person detection and foreground extraction.

	Walk	Pickup1	Pickup2
Walk	78.28%(1074)	20.04%(275)	1.68%(23)
Pickup1	1.86%(21)	84.64%(953)	13.5%(152)
Pickup2	3.30%(39)	11.51%(136)	85.19%(1007)

Table 1: Frame-by-frame confusion matrix on the Grocery Dataset

Our system has an accuracy of ≈82.5% for *frame-by-frame* classification, without any temporal information. It is difficult to show a comparison with [21], as the authors do not report per-frame classification results. Our dominant action classification over individual segments of video is perfect (as also in [21]), where each segment contains a single actor performing a single action. The per-frame classification performance can be significantly improved by including temporal reasoning into our system. Further, we also note that these results were obtained using models trained on completely different dataset of similar actions collected in a different setting which demonstrates the transferability of our approach.

8. CONCLUSION

We presented an approach to track a person through changing poses in realistic indoor settings using a *CRF-Filter*. Our approach combines shape based pedestrian detectors and foreground blobs to obtain an accurate bounding box around the person. We then used this as input to our action recognition system, for frame-by-frame classification using key pose matching. In addition, we also presented a simple approach to learn motion models without accurate camera calibration. We tested our approach in indoor settings, and presented results on a dataset collected in a grocery store. Our entire system runs at 5fps on a standard PC, and produces ≈82.5% with models trained on a different dataset.

9. REFERENCES

[1] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *ICCV*, pages 1395–1402, 2005.

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

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

[4] M. Bregonzio, S. Gong, and T. Xiang. Recognising action as clouds of space-time interest points. In *CVPR*, pages 1948–1955, 2009.

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR (1)*, pages 886–893, 2005.

[6] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, 2005.

[7] 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.

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

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

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

[11] N. Ikizler and D. A. Forsyth. Searching video for complex activities with finite state models. In *CVPR*, 2007.

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

[13] I. Laptev. On space-time interest points. *IJCV*, 64(2-3):107–123, 2005.

[14] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

[15] Y. Li, C. Huang, and R. Nevatia. Learning to associate: Hybridboosted multi-target tracker for crowded scene. In *CVPR*, pages 2953–2960, 2009.

[16] B. Limketkai, D. Fox, and L. Liao. Crf-filters: Discriminative particle filters for sequential state estimation. In *ICRA*, pages 3142–3147, 2007.

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

[18] K. Mikolajczyk and H. Uemura. Action recognition with motion-appearance vocabulary forest. In *CVPR*, 2008.

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

[20] P. Natarajan and R. Nevatia. View and scale invariant action recognition using multiview shape-flow models. In *CVPR*, 2008 .

[21] P. Natarajan, V. K. Singh, and R. Nevatia. Learning 3d action models from a few 2d videos for view invariant action recognition. In *CVPR*, 2010.

[22] J. C. Niebles and F.-F. Li. A hierarchical model of shape and appearance for human action classification. In *CVPR*, 2007.

[23] V. Shet, S. N. Prasad, A. Elgammal, Y. Yacoob, and L. Davis. Multi-cue exemplar-based nonparametric model for gesture recognition. In *ICVGIP*, 2004.

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

[25] T. Starner, J. Weaver, and A. Pentland. Real-time american sign language recognition using desk and wearable computer based video. *PAMI*, 20(12):1371–1375, 1998.

[26] C. Vogler and D. Metaxas. Parallel hidden markov models for american sign language recognition. *ICCV*, pages 116–122, 1999.

[27] D. Weinland, R. Ronfard, and E. Boyer. Automatic discovery of action taxonomies from multiple views. In *CVPR*, pages II: 1639–1645, 2006.

[28] 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.