Human Action Recognition using a Dynamic Bayesian Action Network with 2D Part Models

Vivek Kumar Singh*
University of Southern California
Los Angeles, CA 90089, USA
viveksin@usc.com

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

ABSTRACT

This paper presents an approach to simultaneously track the pose and recognize human actions in a video. This is achieved by combining Dynamic Bayesian Action Network (DBAN) with 2D body part models. Existing DBAN implementation relies on fairly weak observation features which affects recognition accuracy. In this work, we propose to use an occlusion sensitive 2D body part model for accurate pose alignment, which in turn improves both pose estimate and action recognition accuracy. To compensate for the additional time required for alignment, we use an action entropy based scheme to determine the minimum number of states to be maintained in each frame while avoiding sample impoverishment. We demonstrate our approach on a hand gesture dataset with 500 action sequences, and show that compared to DBAN, our algorithm achieves 6% improvement in accuracy.

Keywords

Human action recognition, Dynamic Bayesian Network, Pictorial structure

1. INTRODUCTION

The objective of this work is to recognize single actor human actions in videos captured from a single camera. Automatic human action recognition has a wide range of applications including human-computer interaction (HCI), visual surveillance and automatic video retrieval and has been a topic of active research in computer vision. Existing approaches differ on how the actions are modeled and how they are matched to the observations. In this work, we represent the actions as a sequence of simple action primitives represented in a Dynamic Bayesian Network (DBN), referred to as a Dynamic Bayesian Action Network (DBAN). Most likely activity sequences of actions, based on observations are computed from the DBAN. Observations are derived from the shape of the extracted foreground blobs corresponding to human actors. Our work follows closely the approach described in [11] but that work uses only the overall shape of the blobs as descriptors whereas we incorporate a more elaborate part-based analysis and show resulting improvements in performance.

A popular approach to recognize human actions is to use histograms of sparse spatio-temporal features [7] and use a classifier (such as SVM) to determine the action label. Such approaches are attractive in that because no explicit action modeling is required but they require a large amount of training data to capture viewpoint and other variations; they are also difficult to apply to the task of continuous action recognition. An alternative approach is to use graphical models to represent the evolution of the actor state in a video, for e.g. using HMMs [12], DBNs [8, 11] and CRFs [14, 10, 9]. The actor state is generally represented using a human model with 3D joint positions [8], 2D part templates [5] or an implicit representation using latent variable models [4, 16]. Learning these models requires motion capture data which can be difficult to collect.

Recently, [11] proposed a method to learn Dynamic Bayesian Action Network (DBAN) models from a small number of 2-D videos. This method computes the likelihood of a sampled pose by matching the foreground feature vectors computed over the projected human model with that obtained from the observed image. Simple pose matching metrics, such as foreground overlap [17, 14], have been popular in human action recognition literature due to their efficiency but are sensitive to foreground noise. Further note that the matching is not straightforward, since the person scale and shape variations across different actors must also be taken into account. While local descriptors such as Shape Context [8] can be used for robust matching across shape variations, they are sensitive to small variations in blob shape and computing these descriptors is also computationally expensive. Another commonly used feature is optical flow [6, 10, 2] but obtained flows can be extremely noisy.

We propose to use an intermediate 2D body part representation of the human model to accurately match the human model and image observations across shape variations and observation noise. We refer to the extended DBAN model as *DBAN-Parts*. Given a person scale and approximate viewpoint, the 3D pose is orthographically projected to 2D to determine the visible parts. A 2D part based model (pictorial structure [3]) is then used over the visible parts to accurately align the 2D pose using belief propagation. The likelihood of the pose to recognize human actions is then

*Corresponding author

ICVGIP '10, December 12-15, 2010, Chennai, India

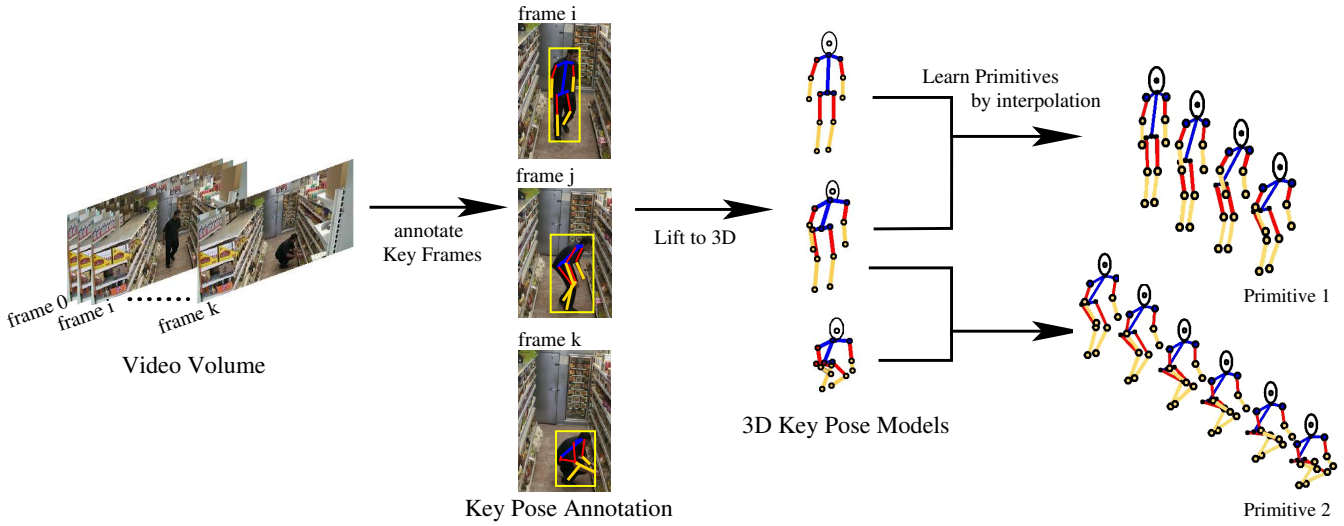

Figure 1: Action Model Illustration for *Crouch* action with 3 keyposes and 2 primitives (figure obtained from [11])

computed over the aligned parts. This intermediate step allows for efficient and more accurate local search for matching the 3D model to the image observation, resulting in more accurate action recognition. Furthermore, fewer samples need to be maintained which partially compensate for the time required to compute pose alignment. To further speed up the inference, we automatically determine the minimal number of hypotheses to be maintained at each step by defining an action class entropy.

To distinguish our method from that of [11], we define the following terminology: method used in [11] will be called DBAN-FGM (FGM standing for foreground matching) whereas the method used in this paper will be called DBAN-Parts. The proposed DBAN-Parts differs from DBAN-FGM of [11] by addition of two key modules:

1. Using 2D Part Model for accurate alignment by performing local shape and position search for each part
2. Sample selection scheme based on action uncertainty to automatically determine the appropriate number of samples require to represent the current state distribution

While DBAN-FGM [11] performs well on datasets with large pose variations such as Grocery Store and Weizmann set [2, 11], the recognition accuracy on the gesture dataset with subtle pose variations is quite low, especially with less training data; the Gesture dataset [11] has about 500 segmented action sequences with a variety of arm gestures common in HCI applications. In this work, we show results on the Gesture dataset and demonstrate that using the 2D part model to compute the pose likelihood allows for a more accurate action recognition and pose estimation.

In the rest of the paper, we first discuss action model representation in section 2; we also briefly describe how the actions are mapped to a Dynamic Bayesian Action Network. Next, we present a modified inference algorithm for efficient pose tracking and action recognition over DBAN in section 3, followed by the part model representation and alignment in section 4, results in section 5 and conclusion in section 6.

2. ACTION REPRESENTATION

Our action representation is based on the concept that a *composite* action can be decomposed into a sequence of simple *primitive* actions. Each primitive action *pe* modifies the *state* s of the actor to give a new state s'. For example, we consider *walking* as a composite action that involves four primitives - left leg forward → right leg crosses left leg → right leg forward → left leg crosses right leg. Each primitive can be defined as a conjunction of rotation of body parts, for e.g. during walking, rotation of upper leg about the hip and rotation of lower leg about the knee. Figure 1 shows an illustration of action model obtained for crouching action (figure obtained from [11]).

Such representations can be obtained either from 2D pose and action boundary annotations [11, 15] or from 3D Motion Capture sequence of the action [8], if available. To obtain such a representation for each composite action, we first manually select the keyposes for each action; each keypose marks a discontinuity in the angular representation of the human pose. We then obtain the 3D model for each keypose, either from lifting 3D pose from 2D annotations [11, 15]; alternatively if the MoCAP is available one may obtain the keyposes by computing pose energy [8].

In this work, we use 2D annotation approach as it doesn't rely on the availability of MoCAP data for all the actions (see [15, 11] for details on lifting 3D human pose from 2D annotations). At this stage, each composite action is essentially a sequence of 3D keyposes with time intervals. Now for every consecutive keypose pair, we define the primitive as the per time step transformation required to go from one keypose to the next i.e. the primitive transforms one keypose to next over a time duration. This duration model allows us to model the speed variations across multiple actors. Now since during a primitive, each part has rotated about a single axis, each primitive can be simply defined as a conjunction of the rotation of body parts. Note that using this representation, we can obtain a strong prior on the 3D pose of a person performing a composite action, after time t has elapsed from the start of the action.

2.1 Dynamic Bayesian Action Network

Given the action models in the form of parametric functions, f, [11] embeds them into a *Dynamic Bayesian Network (DBN)* which is refered to as the *Dynamic Bayesian Action Network (DBAN)*. DBAN used in [11] correspond to the first 3 layers on the model shown in Figure 2, with foreground observation nodes (not drawn in the figure for clarity). The nodes in the topmost layer in the DBAN correspond to the composite actions like *walk*, *flap*, etc. The second layer corresponds to the primitives and the third layer corresponds to the human pose. A duration node is associated with each primitive that captures the time elapsed (in number of frames) since the primitive started. Thus, the *state* s_t of the DBAN at time t is denoted by the tuple (ce_t, pe_t, d_t, p_t) . In this work, instead of directly evaluating the projection of 3D pose on the observations as in [11], we represent the projected pose using a 2D part model, which is represented as the fourth layer in Figure 2.

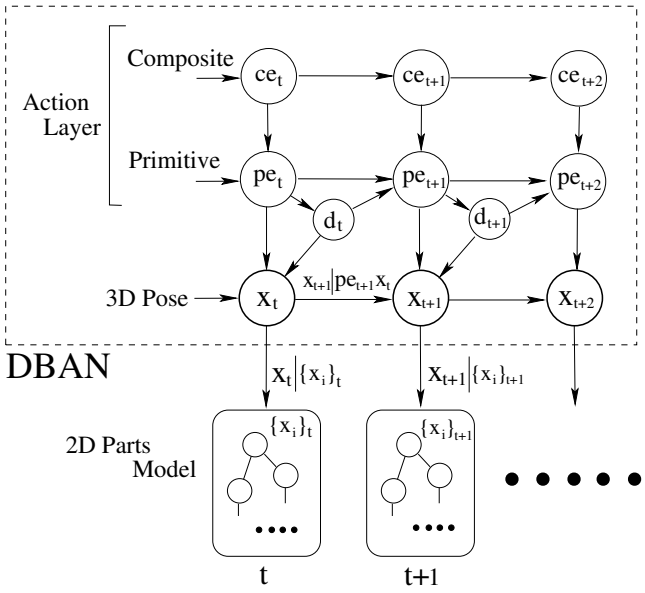

Figure 2: Dynamic Bayesian Action Network with 2D Part Model (DBAN-Parts); DBAN is enclosed within the dotted box. Observation nodes are not shown for clarity.

The optimal state sequence $s^*_{[1:T]}$ for an observation sequence of length T is computed by maximizing the weighted sum of potentials similar to [1, 11].

$$s^*_{[1:T]} = arg \max_{\forall s_{[1:T]}} \sum_{t=1}^{T} \sum_{f} w_f \phi_f(s_{t-1}, s_t, I_t) \qquad (1)$$

where, $\phi_i(s_{t-1}, s_t, I_t)$ are observation and transition potentials and w_i is the weight vector that models the relative importance of the potential functions. Note that DBAN is a multi-variable state representation of the HMM in [1]. Further note that the objective function given by equation 1 has the same form in both DBAN-FGM and DBAN-Parts, however, the observation potentials are different since DBAN-Parts uses part models.

2.1.1 Transition Potential

In this work, we use transition models used in [11]. The primitive transitions are modeled by using the primitive event durations in the *log* of *signum* function, such that the probability of staying the same primitive pe_t decreases near the mean duration $\mu(pe_t)$ and the probability of transition to a new primitive increases.

The pose transition potential is modeled using normal distribution with the mean and variance $\mathcal{N}(\theta_{mean}, \theta_{var})$ of displacement of body joints learnt during the training.

2.1.2 Observation Potential

The observation potentials of a state $\phi_{obs}(s_t, o_t)$ using features we extract from the video. DBAN-FGM projects the 3D pose and computes the likelihood of the projected pose using multiple features, such as foreground overlap and difference image match. In this work, however, we project 3D pose to obtain a 2D part model and which allows an efficient local search and more accurate fit to the observation. For completeness, below we describe te features used in [11] that are relevant to our experiments.

Foreground Overlap: The foreground overlap score of the pose p is computed by accumulating the foreground pixels overlapping with orthographic projection of pose p.

Difference Image Matching: This measures the similarity between the instantaneous motion observed in the video, and the pose change. An estimate of instantaneous changes in the observed pose is obtained as the difference between the frames I_t and I_{t-1} within the person detection window. The observation potential of state s for change in pose is then modeled as the overlap between the difference image and projection of moving body parts on the image.

2.1.3 Relative Weight Vector

In DBAN, feature weight estimation is formulated as the minimization of the log likelihood error function over the entire training set $\mathcal{T}$. Due to the log linear formulation of the likelihood error function, *Voted Perceptron algorithm* [1] can be used to efficiently solve the minimization problem. However to avoid pose annotations in all frames, [11] introduced *Latent State Voted Perceptron algorithm* that deals with missing data. The training algorithm takes M passes over the training set. For each training sequence, the most likely state sequence with the current weight vector is computed. If the estimated composite event is not correct, ground truth state sequence is estimated from the labeled event sequence without the action prediction step (since the action is known). The feature errors between the observed and the ground truth sequence are collected over the entire training set and is used to update the weight vector. For details, please refer to [11].

3. SIMULTANEOUS POSE TRACKING AND ACTION RECOGNITION

Here, we describe the algorithm to simultaneously track human pose and recognize the action in a video using DBAN-Parts. DBAN-FGM [11] infers the action label by matching all action models with observation sequence and finding the best match. Matching is done by sampling poses from action models and fitting the model to the observed image. For ef-

ficiency, instead of matching each action model separately and then selecting the best match, all models are matching simultaneously in one-pass by maintaining multiple state sequences. Since the number of possible sequences is combinatorial, all possible sequences can not be considered. DBAN-FGM [11] uses a greedy strategy and maintains top N state sequences that have the highest score. This greedy selection step is too aggressive. If the number of samples N is small it often results in impoverishment of state samples from all actions thereby leaving samples only from one action after just a few frames; once all the samples from an action are pruned out, that action class is never reconsidered. If however the number of samples is too high, it drastically slows down the inference; furthermore, if after a few frames all state samples belong to the same action class, maintaining large number of samples as no benefit on accuracy and only hurts due to high computational cost.

In this work, we estimate the appropriate number of samples that needs to be maintained at each frame such that state samples from all likely actions are well represented; this is done by defining a measure of uncertainty over the currently active action labels. Below, we present the step-by-step description of the proposed inference algorithm using DBAN-Parts. The key difference between the DBAN-FGM [11] and DBAN-Parts is the use of 2D Part Models and uncertainty based sample set selection, which are described in subsections 3.3 and 3.4 respectively. The pseudo-code for the proposed algorithm is shown in Algorithm 1.

Algorithm 1 Inference Algorithm

▷ Obtain initial states by sampling poses from all composite action models $S_1=\{\langle s_0^{(i)},\alpha_0^{(i)}\rangle | i = 1...N_{max}\}$
for $t = 0$ to T **do**
  ▷ Obtain observation feature maps O_{t+1}
  for all $s_t^{(i)}$ **do**
    ▷ $d_{t+1} = d_t + 1$
    ▷ Obtain $\langle ce_{t+1}^{(i)},pe_{t+1}^{(i)}\rangle \leftarrow \text{allow}(ce_t, pe_t, d_{t+1})$
    for all $\langle ce_{t+1}^{(i)},pe_{t+1}^{(i)}\rangle$ **do**
      ▷ Sample pose from the action model
        $p_{t+1}^{(i)} \sim \phi_p(p_t, pe_{t+1}, p_{t+1})$
      ▷ Compute the state potential α,
        $\alpha_{t+1}^{(i)} = \alpha_t^{(i)} + \sum_f w_f \phi_f(s_t^{(i)}, s_{t+1}, o_{t+1})$
      ▷ Push $\langle s_{t+1}, \alpha\rangle$ to S_{t+1}
    end for
  end for
  ▷ Obtain action class likelihood vector, v
    $v = \{v_{ce}\}$, where $v_{ce} = \max_{s_{t+1}^{(i)}=\langle ce,\ldots\rangle} \alpha_{t+1}^{(i)}$
  ▷ Set target sample set size,
    $N_t \propto \left(\sum_{ce} v_{ce} log(v_{ce})\right) \times N_{max}$
  ▷ Prune S_{t+1} such that $|S_{t+1}| \leftarrow \max(N_t, N_{min})$
end for
▷ actionlabel $= \arg\max_{s_T^{(i)}} \alpha_T^{(i)}$

3.1 Initialization

For initializing the state distribution, we sample poses from all the composite actions in the action set. For viewpoint invariance, all likely viewpoints are considered for each pose sample from every composite action model.

3.2 Prediction: Sample Next State

For each state s_t, we increment the duration of the state by unit time step. Given the current action (ce_t, pe_t) and new duration, we then sample the next action state (ce_{t+1}, pe_{t+1}). Note that if primitive transition occurs, then the duration is set to 0 to mark the start of a new primitive. Next, we sample from the *pose transition potential* $\phi_p(p_t, pe_{t+1}, p_{t+1})$ to choose the next pose p_{t+1}. Here, the pose transition potential represents a distribution over the parameters α in the function $f_{pe_{t+1}}(p, p', \alpha)$ corresponding to primitive pe_{t+1}.

3.3 Fit the Sampled State to the Observation

We first apply a pedestrian detector [18] to find the person in the video, thus our algorithm initializes only when a standing pose is observed. We then apply a combined shape and foreground blob tracker to track and localize the person in each frame, even through changing poses. The position and scale information available from the person tracker is then used to adjust the 3D pose sampled from the action model in the previous step. Given the adjusted 3D pose, we then orthographically project the pose to construct a 2D part model which is then used for accurate localization. Note that during the projection step, we automatically determine the non-observable/occluded parts and do not use those parts for localization. Figure 4 show some sample 3D poses and corresponding 2D part models. Using the 2D part model, we then perform a local search to accurately fit the pose to the observation. This local search allows us to handle noise better than in DBAN-FGM. The details on obtaining the 2D part model from 3D pose and the local search is described in detail in Section 4.

The likelihood of the pose/state is then computed by matching the localized 2D pose with low-level image features. This includes computing the observation potential $\phi_{obs}(s_{t+1}, o_{t+1})$ using foreground match, difference image and part templates (described later in Section 4.2).

3.4 Selecting the State Samples

Since maintaining all possible state sequences is not possible, only a small number of states are retained in each frame. As discussed earlier, a greedy sample selection step can lead to sample impoverishment and may significantly affect both accuracy and efficiency of the algorithm. To avoid action sample impoverishment, we set the minimum number samples N_{min} to be maintained in each frame; we also set N_{max}^a as the maximum number of samples allowed for any action class. Note that this may address the sample impoverishment from different action classes but still has poor efficiency.

We define a measure of action label uncertainty in the current frame by computing the entropy over the distribution of currently active actions/states; an action is considered *active*, if there is a state sample corresponding to that action. To compute the entropy of the currently active actions, we compute the action class likelihood vector $v = \{v_{ce}\}$, where v_{ce} is the highest likelihood score over all states in the current frame that belong to the action class ce. Given the action class likelihood vector v, we then define the target

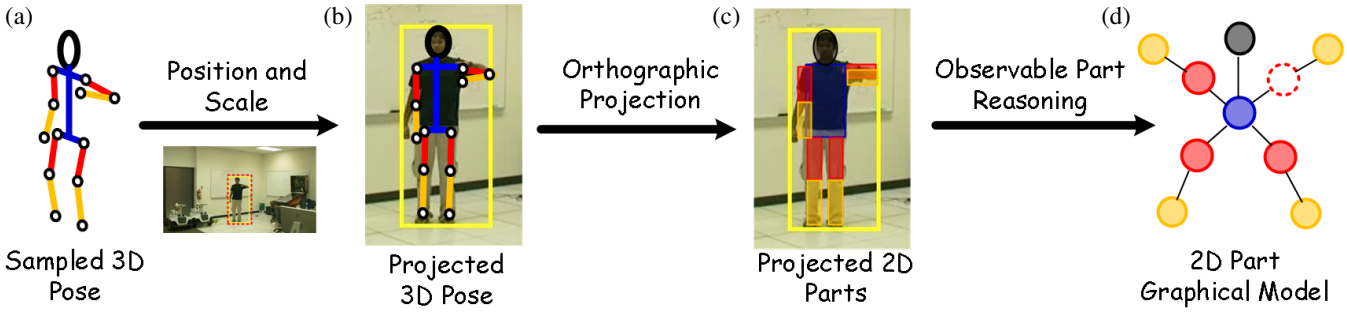

Figure 3: Illustration of estimating 2D Body part models from the 3D Pose; Note that the node corresponding to right upper arm is non-observable (due to occlusion) and is marked with dotted boundary in (d)

sample set size N_t in the current frame t as,

$$N_t \propto \left(\sum_{ce} v_{ce} log(v_{ce}) \right) \times N_{max}$$

Note that when the uncertainty is high, large number of samples are maintained thereby allowing presence of samples from multiple action classes and avoiding impoverishment. When uncertainty is low, the samples are likely to the belong to the same/few action class, and thus only a few samples are enough for accurate inference; note that maintaining fewer samples also speeds up the inference.

4. 3D POSE OBSERVATION USING 2D BODY PART MODEL

In this section, we describe the localization of a 3D pose projected from a given viewpoint. This is achieved using a graphical model of the 2D body parts. The body part model used in the work is similar to the *Pictorial Structures* [3] which is widely used for estimating human pose in an image. The model has 10 nodes, each corresponding to a body part - head, torso, upper arms (l, r), lower arms (l, r), upper legs (l, r) and lower legs (l, r). These nodes are connected with edges that capture the kinematic relationship between the parts. Figures 4 and 3 show the part model. For localization using *Pictorial structure*, the individual parts are searched by applying part detectors at all locations and orientations followed by belief propagation to enforce kinematic constraints. Note however that in this work, our objective is to accurately localize a 3D pose projected from a given viewpoint. This imposes a very strong constraint on the orientation of the body parts and their kinematic relationship. Furthermore, approximate position and scale information is also available from the person detection step. Hence, localization in our case does not require a dense part search. However, during localization, we need to tackle the problem that some of the body parts may not be observable, either due to inter-part occlusion (see Figure 3) or 3D-2D projection (see Figure 4(b)).

Now we first describe the process to generate the 2D part model appropriate for localization, and then we briefly describe the localization using local search.

4.1 Building 2D Part Models

We project the 3D pose from the given viewpoint to estimate the 2D position of the body joints. From the body

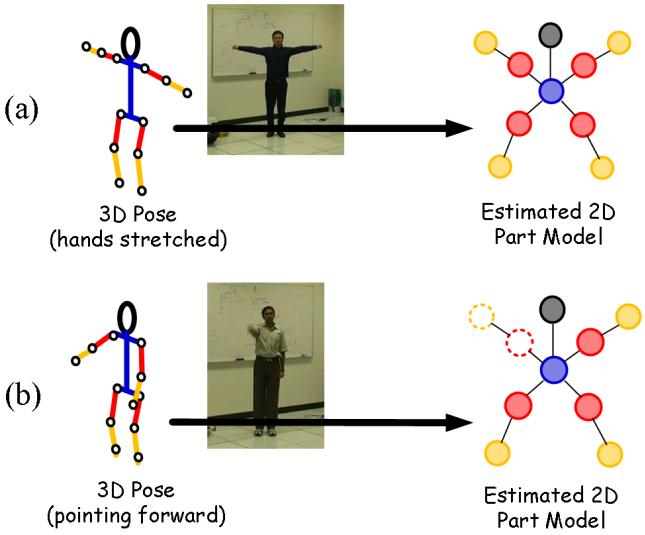

Figure 4: Sample 3D poses and their corresponding 2D part models used for alignment; non-observable nodes in (b) are marked with dotted boundary.

joints, we build a rectangular cardboard model by fitting a rectangle between every pair of joints connected by a body part (shown in Figure 3(c)). During projection, we also estimate the relative depth order of the 2D parts. Next, we determine which parts are visible based on the depth order and pairwise overlap between the part rectangles. In our experiments, we considered parts with percentage visibility below 50% to be occluded. Furthermore, when projecting 3D pose to 2D, some of the body parts are too small to be observed (see Figure 4(b)) and thus are not useful for localization. Figure 3 shows the flowchart of building the 2D part model for a 3D pose from a given viewpoint.

4.2 Part Detectors

We use template based matching for detecting body parts in the image. We model the head with an ellipse template, torso with an oriented rectangle and each arm with a foreshortened pair of lines. We define the likelihood score $\phi(x)$ of a part hypothesis x using both the strength and orientation

of the gradient at each point in the model.

$$\phi_{edge}(x) = \sum_{x_i \in x} d_{mag}(I(x_i)) \times d_{ori}(x_i, I(x_i))$$

where, $d_{mag}(I(p))$ is the approximate Euclidean distance to the nearest edge pixel from the image point p, weighted by the edge strength. This can be calculated very efficiently using generalized distance transform over the edge likelihood map [3]; $d_{ori}(p, I(p))$ is the orientation likelihood, which is the dot product between the normals at the model point p and corresponding point in the image $I(p)$. Since the orientation information is often very noisy, we approximate the normals by quantizing into 8 bin orientations.

4.3 Localization using 2D Part Models

Given the scale adjusted 3D pose X and position information, we apply an edge template for each part x_i over the expected configuration of the part and a small neighborhood around it; configuration of a part is given by its position, scale and 2D orientation. We then enforce kinematic constraints between the obtained part distributions using message passing (similar to Pictorial Structure [3]). The posterior likelihood of the full pose X given the image observation O is given by,

$$\mathcal{F}(X_p, Y) = \sum_{i \in V} \phi_i(y_i|x_i) + \sum_{ij \in E} \psi_{ij}(x_i, x_j) \quad (2)$$

where V is the set of all body parts, E is the set of part pairs that are kinematically connected, and $y_i \in Y$ is the likelihood map of part i. The best aligned 2D pose is obtained by maximizing the posterior likelihood $\mathcal{F}(X_p, Y)$. Note that occlusion sensitive pose localization proposed here is different from that used proposed in [13]. Compared to [13] which consider pixel level occlusion constraints for each part, we only consider parts which are almost completely visible (with visibility more than 80%). This allows our inference to be much more efficient and as we show in our results, the localization is accurate enough for reliable action recognition.

5. EXPERIMENTS

We tested our approach on the Gesture Dataset [11]; the dataset includes about 500 instances of hand gestures used in HCI applications.

[Dataset Description]
The gesture dataset has 12 actions performed by 8 different actors, captured from a static camera in an indoor lab setting. The action set include - Column Left (bend left arm from side to overhead), Column Right, Open Up (move both arms from overhead to side), Close Up (move both arms upward from side), Turn Right (extend arm to right side), Turn Left, Line (extend arms parallel to ground), Close Distance (clap), Stop Right (raise right arm upward to the full extent of arm), Stop Left, Action Left (extend both arms, then raise left arm overhead), Action Right. Figure 5 shows sample frames from the dataset.

Each action sequence in the dataset has exactly one person performing the action, facing the camera. Each actor performs every action about 5 times, thus the dataset contains a total of about 500 action sequences. The videos are 852×480 resolution, and the height of person varies between 200-250 *pixels* across different actors. This set is similar to that used in [12] but has a bigger variety. As the background is not highly cluttered, extracted foreground is quite accurate but the large number of actions with subtle differences makes recognition still a challenging task.

[Experiment Settings]
To compare our inference algorithm with that in [11], we use the same experiment settings wherever possible. The models for each action were obtained by video annotation. For learning the feature weights for each action model, the same training data on which the action model is trained. The feature weights were randomly initialized and the one that achieves the highest accuracy on the training set was used during testing.

During inference, we set the minimum samples for each action N_{min} to 3 and maximum number of samples in any frame N_{max} to 15. In our experiments, the actual number of samples in a frame varied between 3 and 15 due to the entropy based sample set selection, and on an average about 7 samples were maintained in each frame.

[Quantitative Evaluation]
To evaluate the performance of our approach, we computed both the action classification accuracy and the error in pose estimates. We split the action sequences into train and test sets based on the actors i.e. the action models trained on a subset of actors and test on the rest. Since each video sequence contains only one action; it is said to be recognized correctly if its label is the same as in the ground truth. Since the primary contribution of DBAN-FGM [11] is on learning action models with low training requirements, we ran our experiments with train:test ratio of 1:7. Second column of the Table 1 provides the recognition results, averaged over 2 training sets. The accuracy numbers for 1:7 train:test for DBAN-FGM were obtained from the Figure 6(b) in [11].

To evaluate the effect of our 2D Part model, we evaluate the errors in the pose estimate obtained using our inference with DBAN-FGM (that does not use 2D part model). To measure the error in pose estimates, we manually annotated 48 2D poses with 4 randomly selected frames from an instance of each action class, and compute the accuracy of the estimated 2D parts. A 2D part estimate is considered correct if it lies within the length of the ground truth segment. Since our experiments are on hand gestures, a more meaningful evaluation is to compute the pose accuracy only over the arms, since arms are the only parts involved in the action. Third column of the Table 1 provides the pose accuracy computed over the arms (192 annotations); the number within parentheses show the accuracy over all the body parts (480 annotations). Even though the improvement in pose accuracy averaged over all parts is not quite significant (only 2.5%), notice that the accuracy over the parts relevant to the action (arms) is about 6%.

Approach	Train:Test ratio	Recognition (% accuracy)	2D Tracking (% accuracy)
DBAN-FGM [11]	1 : 7	78.6	75.67(89.94)
DBAN-Parts	1 : 7	**84.52**	**81.76**(92.66)

Table 1: Performance on Gesture Store dataset

We also report the confusion matrix for the recognizing actions using DBAN-Parts over the entire dataset. Figure 6 shows the confusion matrix for train:test ratio of 1 :

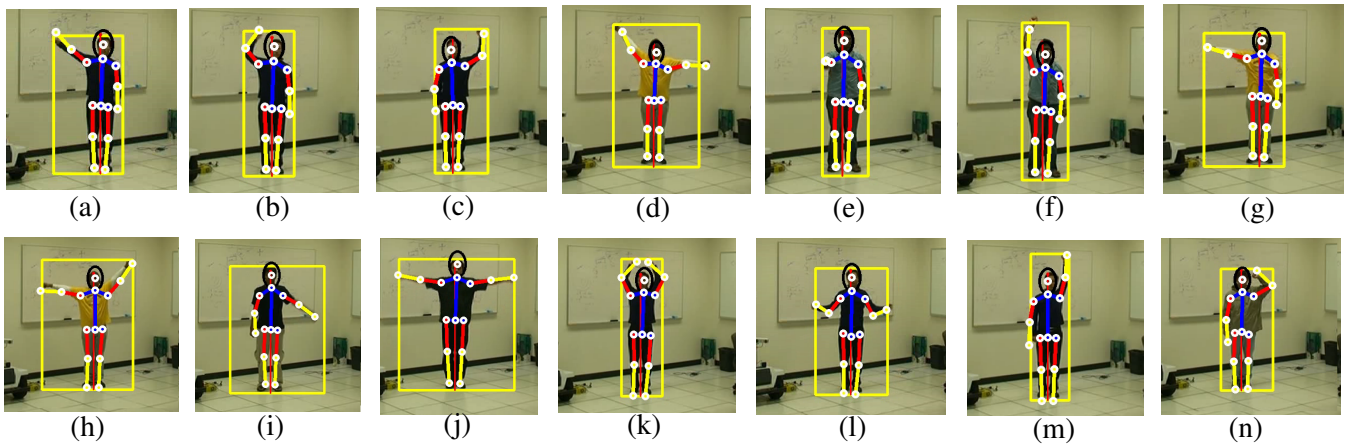

Figure 5: Results on the Gesture Dataset: The bounding box shows the person position and the estimated pose is overlaid on top of the image, illustrated by limb axes and joints.

7. Notice that the recognition accuracy is around 85 − 90% for each action, except for *Line* and *OpenUp* actions which got misclassified as *TurnRight* and *ColumnRight* respectively. Observe that in both cases, action model of one arm is same and hence the confusion is expected due to similarity in poses. We believe more accurate part detectors would be able to localize better and deal with such ambiguities. Further note that the actions (*CloseDistance*, *StopRight* and *StopLeft*) that contain poses where arms are non-observable/occluded, get correctly recognized; Figure 5(e) shows an example of such a pose in *StopRight* action, where right arm is occluded.

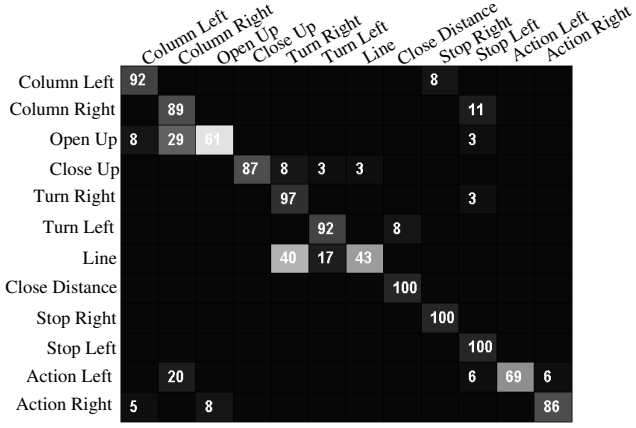

Figure 6: Confusion matrix for the Gesture Dataset

6. CONCLUSION

In this work, we have presented a general framework for simultaneous tracking and action recognition using 2D part models with Dynamic Bayesian Action Network [11]. The 2D part model allows more accurate pose alignment with the observations, thereby improving the recognition accuracy. To compensate for the additional time required for 2D part alignment, we proposed an action entropy based scheme to determine the minimum number of samples to be maintained in each frame while avoiding sample impoverishment. In future, we plan to apply this algorithm in more complex domains with cluttered environments by employing more accurate part detectors and extend this framework to recognize actions that include multiple actors.

7. REFERENCES

[1] M. Collins. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1–8, July 2002.

[2] A. Fathi and G. Mori. Action recognition by learning mid-level motion features. In *Computer Vision and Pattern Recognition (CVPR)*, 2008.

[3] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *International Journal on Computer Vision (IJCV)*, 61(1):55–79, 2005.

[4] A. Gupta, F. Chen, D. Kimber, and L. S. Davis. Context and observation driven latent variable model for human pose estimation. In *Computer Vision and Pattern Recognition (CVPR)*, 2008.

[5] N. Ikizler and D. A. Forsyth. Searching video for complex activities with finite state models. In *Computer Vision and Pattern Recognition (CVPR)*, 2007.

[6] Y. Ke, R. Sukthankar, and M. Hebert. Event detection in crowded videos. In *International Conference on Computer Vision (ICCV)*, 2007.

[7] I. Laptev. On space-time interest points. *International Journal on Computer Vision (IJCV)*, 64(2-3):107–123, 2005.

[8] F. Lv and R. Nevatia. Single view human action recognition using key pose matching and viterbi path searching. In *Computer Vision and Pattern Recognition (CVPR)*, 2007.

[9] L.-P. Morency, A. Quattoni, and T. Darrell. Latent-dynamic discriminative models for continuous gesture recognition. In *Computer Vision and Pattern Recognition (CVPR)*, 2007.

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

[11] 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.

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

[13] L. Sigal and M. J. Black. Measure locally, reason globally: Occlusion-sensitive articulated pose estimation. In *CVPR*, pages 2041–2048, 2006.

[14] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas. Conditional random fields for contextual human motion recognition. In *International Conference on Computer Vision (ICCV)*, pages 1808–1815, 2005 .

[15] C. J. Taylor. Reconstruction of articulated objects from point correspondences in a single uncalibrated image. In *Computer Vision and Image Understanding (CVIU)*, volume 80, pages 349–363, 2000.

[16] R. Urtasun, D. J. Fleet, and P. Fua. 3d people tracking with gaussian process dynamical models. In *Computer Vision and Pattern Recognition (CVPR)*, pages 238–245, 2006.

[17] D. Weinland, R. Ronfard, and E. Boyer. Automatic discovery of action taxonomies from multiple views. In *Computer Vision and Pattern Recognition (CVPR)*, pages II: 1639–1645, 2006.

[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.