Predicting Body Movement and Recognizing Actions: an Integrated Framework for Mutual Benefits

Boyu Wang and Minh Hoai
Stony Brook University, Stony Brook, NY 11794-2424, USA
{boywang, minhhoai}@cs.stonybrook.edu

***Abstract*— Human action recognition and body movement prediction are important tasks. They are different and have traditionally been addressed separately. These tasks, however, provide mutual benefits to each other, and existing methods fail to capture these benefits. In this paper, we propose a method for jointly recognizing the action and predicting the movement of a person. Our method is based on two Long-Short Term Memory (LSTM) recurrent neural networks, but extend them to provide and receive benefits of each other. In particular, we design two LSTM architectures. One LSTM can generate a sequence of body movement conditioned on the past movement and the predicted class of the action, and the other LSTM can recognize the human action based on the predicted sequence of body movement. Experiments on Montalbano and MSR Action 3D datasets show that movement prediction provides benefits to early recognition of human action, which in turn improves the quality of the predicted movement.**

I. INTRODUCTION

The ability to predict human body movement has applications in a wide range fields, ranging from robotics and entertainment to surveillance and health care. For example, consider a future scenario where a companion robot shares the living space with humans. A key requirement for the robot would be its ability to physically interact with humans. The robot must be able to recognize and predict human body movement and react in a timely manner; otherwise, the physical interaction would be slow and unnatural.

Predicting the body movement is different from recognizing the action class of the body movement. The former forecasts the future, while the latter analyzes the current and past observations. Recognizing the classes of human actions is important, but insufficient for real-time human robot interaction. There exist some methods for early recognition of human actions, which attempt to recognize the action class of the body movement before the action is complete, e.g., [7, 12, 13, 18, 26, 27, 33]. However, early recognition cannot substitute body movement prediction either. To see this, consider a scenario where you want to shake hands with a robot. You initiate the action by extending your arm toward the robot. For a fluent interaction, the robot must recognize your handshake action as soon as possible (early recognition) and anticipate where your hand will be so the robot can start extending its hand to the right spot to meet your hand. If the robot cannot predict the location of your hand, it will not be able to move its hand to the right location until you have stopped moving your hand. Thus the interaction would be slow and unnatural, just like how the current generation of robots interact with humans.

Fig. 1: Two important problems for human robot interaction: early recognition of human actions and body movement prediction. We propose to jointly address these two problems, exploiting the mutual benefits.

Nonetheless, movement prediction and early recognition provide mutual benefits to each other. On one hand, being able to predict the body movement of a person allows us to visualize and subsequently recognize an action even before it is complete. On the other hand, the body movement of a person can be predicted with higher precision if the on-going action of the human can be recognized.

In this paper, we propose a novel method for joint movement prediction and early recognition. Our method is based on the Recurrent Neural Network (RNN), but extend it to integrate multiple systems. In particular, we combine two LSTM RNNs [16], one for movement prediction and one for early recognition. The LSTM for movement prediction is designed to accommodate the output of the recognition system, while the recognition LSTM uses the predicted sequence of body movement produced by the movement prediction LSTM.

Experiments on two publicly available datasets Montalbano Gesture dataset [8] and MSR Action 3D dataset [22] demonstrate the benefits of jointly performing movement prediction and early recognition. The integrated system that combines the two LSTMs for early recognition and movement prediction outperforms the individual LSTMs,

This project is partially supported by the National Science Foundation Award IIS-1566248.

978-1-5386-2335-0/18/$31.00 ©2018 IEEE

both in terms of early recognition and movement prediction performance metrics. The integrated system also outperforms several state-of-the-art methods which were specifically designed for early recognition.

One contribution of our paper is the development of an integrated framework for movement prediction and early recognition, yielding synergy from their mutual benefits. The proposed framework assumes the availability of 3D skeleton data. The framework consists of two LSTM networks, and the input to each LSTM network is a sequence of 3D skeleton vectors instead of RGB images. While this particular framework is developed specifically for 3D skeleton data, the mutual benefits of movement prediction and early recognition toward each other exists independently of the data representation. Extending beyond skeleton-based representation, our contributions of this paper are:

1) We provide justification and empirical evidence showing that movement prediction is beneficial for early recognition. This has never been considered and demonstrated in the literature of early recognition.
2) We show that early recognition provides benefits for movement prediction. This has also never been considered before.

II. Related Work

In the literature of computer vision, many computational models have been developed for human action recognition, but most of them focus on improving the accuracy of offline processing rather than the timeliness of the decision making (e.g., [14, 15, 29, 31, 36, 37]). Only in recent years, has there been some effort to address early recognition problem [1, 7, 12, 13, 18, 26, 27, 33]. However, none of these works, including our prior attempt [12, 13], studied the benefits of movement prediction. They instead studied the benefits of different feature encodings and classifiers for early recognition. Ryoo [27] proposed integral and dynamic bag-of-word models for early recognition of human interaction. Raptis and Sigal [26] used structured-output SVM to learn the set of most discriminative keyframes for early recognition. Kitani et al. [18] proposed a Markov decision process to obtain a distribution over possible human navigation trajectories. Vondrick et al. [33] learned to anticipate the feature vectors of future video frames by exploiting the temporal order of video frames. Ellis et al. [7] presented a low latency algorithm that could determine distinctive canonical human poses. Zanfir et al. [41] proposed a non-parametric moving pose framework for low-latency human action and activity recognition. The moving pose descriptor considers both pose information as well as differential quantities (speed and acceleration) of the human body joints within a short time window around the current frame. Aliakbarian et al. [1] proposed a novel loss for training the classifier for early recognition. However, none of the aforementioned methods considered the benefits of movement prediction for early recognition.

In the field of robotics, human action forecasting and anticipation is an emerging research area. Koppula and Saxena [19] presented an anticipatory temporal conditional random field to model the distribution of future human activities, which could improve detection accuracy of past activities and enable an assistive robot to plan ahead for reactive responses. Jain et al. [17] proposed to combine spatio-temporal graph with recurrent neural network to generate future human movement. Inverse reinforcement learning was used in [5, 18, 20, 43] to obtain a distribution over possible human navigation trajectories from visual data. [20, 43] modeled the forthcoming interactions with pedestrians for mobile robots. Dragan and Srinivasa [5] predicted the future goals for grasping an object. Wang et al. [38] proposed a latent variable model for inferring unknown human action. The aforementioned methods differ from ours in the way partial actions are modeled and recognized. These methods aim to predict the trajectory or destination of a human subject, and they are only suitable for predicting actions/activities that can be determined by the trajectory or destination of the subject. In this paper, we aim to go beyond the forecasting and classification of the planar or 3D trajectories.

Analyzing the dynamics of human motion is an important research topic, but most prior studies did not explore the benefits from early recognition. Wang et al. [35] used Gaussian process dynamical models for nonlinear time series analysis, which comprised of a low-dimensional latent space with associated dynamics, as well as a map from the latent space to an observation space. Brand and Hertzmann [2] learned distinc motion patterns from motion captured sequence, which could then be used to synthesize novel motion data. These methods required knowing the class label in advance to generate a motion sequence, while our prediction network can work with or without knowing the class label. Qi et al. [25] used a spatial-temporal graph to model the relationship between actions and objects. Future actions are predicted using temporal grammar and the Earley parsing algorithm. Cao and Nevatia [3] estimated poses and motions through analyzing forces. The motion was forecasted by utilizing joint forces to determine joint accelerations and integrating them for the 3D pose locations in all the other frames. This method did not consider long term dependency for prediction. Our approach use the LSTM networks, which have memory cells to store long and short term memory for prediction and recognition. Fragkiadaki et al. [10] proposed Encoder-Recurrent-Decoder model for motion capture generation. Compared to our work, the prediction network cannot predict body movement based on different class label. Similarly, Walker et al. [34] used the variational autoencoder to predict pose at next time and applied conditional generative adversarial network to generate RGB video frames. Pavlovic et al. [24] learned models of human dynamics using switching linear dynamic system models. However, these methods were used for segmentation and offline recognition rather than for motion prediction and early recognition. [9] focused on forecasting sport activities of an adversarial team of players, which is different from human motion prediction. Zeng et al. [42] formulated the prediction problem as the inverse reinforcement learning problem and focused on the

frame level prediction. This method, like many others, did not consider the benefits of early recognition for prediction.

Note that motion prediction is not the only objective of our work. In this paper, we advocate and demonstrate the mutual benefits of motion prediction and early recognition. There might exist a better method for motion prediction than what is being proposed here. However, that method should not be considered as a competitor of our overall approach. A better prediction method can be used to improve our approach, if it can be extended and integrated into our framework for joint motion prediction and early recognition.

III. Movement prediction and action recognition

We propose a framework that jointly performs movement prediction and early recognition of human actions, integrating their complementary benefits. Our framework combines two LSTM recurrent neural networks, one for movement prediction and one for early recognition. In this section, we describe how the two networks are trained and combined. These LSTM networks are designed for processing skeleton data; the input to the LSTM networks is a sequence of 3D skeleton vectors instead of RGB images.

A. Joint Prediction and Recognition

Figure 2 depicts the proposed approach for integrating the benefits of movement prediction and action recognition. In this figure, RegLSTM and PredLSTM are the two recurrent neural networks for action recognition and movement prediction respectively. This method uses both past observed human poses and the predicted poses of the futures to make decisions. At time step t, the input to the recognition LSTM is $\mathbf{x}_t$. $\mathbf{x}_t$ is the actual observed pose $\mathbf{s}_t$ if t is not a future time, and $\mathbf{x}_t$ is the predicted pose $\hat{\mathbf{s}}_t$ if t is in the future (and therefore $\mathbf{s}_t$ has not been observed). The output of the RegLSTM is $\mathbf{p}_t$, a vector of class probabilities. The length of $\mathbf{p}_t$ is the number of the action classes. The input to the PredLSTM consists of both $\mathbf{x}_t$ and the current probability estimates of action classes $\mathbf{p}_t$. Again $\mathbf{x}_t$ is either the observed or predicted human pose, depending on whether t refers to a past or future time step. The output of the prediction network PredLSTM is $\hat{\mathbf{s}}_{t+1}$, which aims to approximate the actual pose at the next time step $\mathbf{s}_{t+1}$. The two LSTM networks are integrated. The predicted poses of the prediction network are used as the inputs to the recognition network, while the class probability vector produced by the recognition network is a part of the input to the prediction network.

The effectiveness of the proposed approach and in particular the necessity of using the predicted poses for early recognition can be justified from a probabilistic perspective. Consider an ongoing action a, we want to estimate the probability of the action a given the sequence of human poses obtained until the current time step t. The probability of action a can be computed by marginalizing over all possible future pose sequences $\mathbf{s}_{t+1:t+\tau}$ with τ being a long-enough

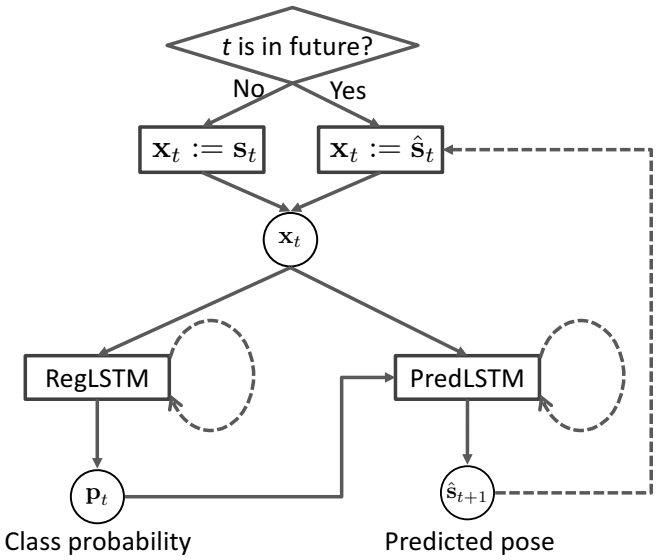


Fig. 2: **Integrative framework for action recognition and movement prediction.** It combines two LSTM recurrent neural networks, RegLSTM for action recognition and PredLSTM for movement prediction. $\mathbf{s}_t$: actual observed human pose at time t, $\hat{\mathbf{s}}_t$ is the predicted pose obtained using the PredLSTM. $\mathbf{p}_t$ is the output of the RegLSTM, which is the probability vector for multiple action classes. Solid arrows indicate the information flow within a time step. Dash arrows are recurrent links that indicate the flow of information between consecutive time steps.

time horizon:

$$P(a|\mathbf{s}_{1:t}) = \int_{\mathbf{s}_{t+1:t+\tau}} P(a|\mathbf{s}_{1:t+\tau}) P(\mathbf{s}_{t+1:t+\tau}|\mathbf{s}_{1:t}) \partial \mathbf{s}_{t+1:t+\tau}.$$

The above equation suggests the importance of learning PredLSTM to approximate $P(\mathbf{s}_{t+1:t+\tau}|\mathbf{s}_{1:t})$. We can compute the probability of the event a by first using PredLSTM to generate samples of plausible future observation sequences, and subsequently compute the marginalized probability over the sample set. This seems counter intuitive because we cannot gain more information than what is given at a time. In principle, a predicted sequence $\hat{\mathbf{s}}_{t+1:t+\tau}$ from $\mathbf{s}_{1:t}$ (using PredLSTM) should not provide more information than the sequence $\mathbf{s}_{1:t}$ itself. However, the benefits of PredLSTM are thanks to the preservation of information rather than the generation of new information. Normally, an action detector is trained to recognize full actions only, and it may not be able to recognize partial actions. This is due to the incorrect focus of modeling effort: an action detector trained to recognize full actions may not pay attention to the characteristic information that occurs at the beginning of the action. Therefore, even though the prediction procedure does not provide additional information, it may preserve useful information that might have been ignored otherwise.

B. Movement Prediction Network

For body movement prediction network, the input vector to the network is the concatenation of skeleton joint coordinates

at time t and the vector that represents the class probability of current sequence. The output vector $\hat{\mathbf{s}}_{t+1}$ represents the prediction of the skeleton at time $t+1$. This network can be used to generate sequences of human movement that are conditioned on a class probability vector. Given the same partial sequence, the network can synthesize different sequences conditioned on different class labels. During training, the class probability vector is from ground truth label. It is a binary vector where the element corresponds to ground truth label has the value of 1 and all other entries are 0. During testing, the probability vector will be the provided by the recognition network RegLSTM.

More precisely, given multiple sequences of human actions that belong to L action classes, we first sample many subsequences of a fixed length T (normally $T = 20$). We train the parameters of the network by minimizing the sum of prediction losses of all training subsequences. Suppose $\mathbf{s}_{1:T}$ is a training subsequence and l is the associated class label. Let $\mathbf{e}_l$ be a binary vector of length L where all entries are 0 except for the l^{th} entry that has the value of 1. The input to the LSTM network at time t is $\mathbf{x}_t = [\mathbf{s}_t; \mathbf{e}_l]$, and the loss for the training subsequence is defined as: $\mathcal{L}_{pred}(\mathbf{s}_{1:T}) = \sum_{t=1}^{T-1} ||\hat{\mathbf{s}}_{t+1} - \mathbf{s}_{t+1}||_2^2$. The derivative of this loss function with respect to the network weights can be efficiently calculated with back-propagation through T time steps. We train the parameters of the LSTM by optimizing the above loss using stochastic gradient descent.

C. Early Recognition Network

For the early recognition network RegLSTM, the input vector $\mathbf{x}_t$ contains the skeleton joint coordinates at time t, and the output $\mathbf{p}_t$ is the predicted class probability vector for the sequence $\mathbf{x}_{1:t}$. The length of the vector $\mathbf{p}_t$ is the number of classes. The training loss for the sequence is based on the negative log likelihood calculated at all time steps. Suppose $\mathbf{s}_{1:T}$ is a training sequence of action class l, the training loss for this sequence is defined as: $\mathcal{L}_{reg}(\mathbf{s}_{1:T}) = -\sum_{t=1}^{T} w_t \log(\mathbf{p}_t(l))$. Here $\mathbf{p}_t(l)$ is the l^{th} entry of the vector $\mathbf{p}_t$. The parameter w_t is the weight for the misclassification penalty at time t. We use different misclassification penalty weights for different time steps to emphasize the importance of recognizing the full action over partial actions. In our experiments, we use $w_t = sigmoid(\alpha(t-\beta))$, where α, β are tunable parameters.

The total training loss is the sum of multiple loss terms, one for each training sequence. This allows the parameters of the network to be trained with stochastic gradient descent. This is an iterative optimization procedure where each iteration updates the network parameters based on the derivatives computed on a batch of training data. The derivatives of the loss of a training sequence with respect to the network parameters can be calculated with backpropagation through time. This requires unrolling the computation graph of the RNN a number of time steps that is equal to the length of the training sequence. However, the lengths of the training sequences are not the same. This requires the network to be unrolled to different lengths. This makes training inefficient because the gradient calculation for multiple training sequences cannot be done at the same time. This problem exists for both CPU and GPU architectures. To overcome this problem and reduce the training time, we use the following procedure. We first group the training sequences with similar lengths together and truncate the sequences in each group to have the same length. We divide training data into batches such that each batch only contains data from a single group, and therefore the training sequences in each batch have the same length. We train an RNN on the truncated sequences and subsequently fine-tune the network using original sequences. Due to the need to compute the gradients sequence by sequence, each epoch of the fine-tuning step takes significantly longer than each epoch of the pre-training step. However, fine-tuning often converges after three or four epochs.

IV. Experiments

We performed experiments on two publicly available datasets. We found that the prediction network provided benefits to early recognition, which in turn improved the quality of prediction.

A. Datasets

We used two publicly available datasets: Montalbano Gesture dataset [8] and MSR Action 3D dataset [22]. The details about these datasets are as follows.

Montalbano Gesture dataset. This dataset was captured with a Microsoft Kinect depth sensor. In all sequences, a camera recored a human subject performing natural communicative gestures. The gesture vocabulary contained 20 Italian cultural/anthropological signs. The gestures were performed by 27 different individuals under diverse conditions. There are 13,858 labeled sequences which contain 1,720,800 frames. Each frame in a sequence contains 20 skeleton joints. The dataset is divided into train, validation, and test subsets, containing 7754, 3362, and 2742 sequences respectively.

MSR Action 3D Dataset. This dataset has 557 valid sequences of 10 subjects performing 20 actions in an unconstrained way. All sequences were captured by a Kinect-like depth sensor. There are about 50 frames in each sequence, and a skeleton has 20 joints. The low accuracy of the estimated 3D poses due to occlusion makes this dataset very challenging. We followed the protocol provided in [22] for train/test split. The data samples of subjects 1, 3, 5, 7, 9 were used for training while the samples of subjects 2, 4, 6, 8, 10 were used for testing.

B. Parameter setting and model training

We centralized the skeleton joints by translating them so that the hip center is at the coordinate origin. Following [6], we reduced the impact of noise by smoothing:

$$\mathbf{f}_t = (-3\mathbf{s}_{t-2} + 12\mathbf{s}_{t-1} + 17\mathbf{s}_t + 12\mathbf{s}_{t+1} - 3\mathbf{s}_{t+2})/35.$$

where $\mathbf{f}_t$ is the smoothed output at time t, $\mathbf{s}_t$ is the raw coordinate values of the skeleton. Furthermore, we normalized

the skeleton data based on the average limb size so that every skeleton roughly has the same average limb size.

For the recognition network RegLSTM, the parameters of the mis-classification penalty weight $\mathbf{w}_t$ are: $\alpha = 0.3$, $\beta = \frac{T}{2}$. For optimization with back propagation through time, we use Adagrad as the optimization method with the learning rate being 0.001.

C. Comparison methods

We compared our proposed action recognition approach with several methods: SVM, Structured-Output SVM (SOSVM) [32], Max-Margin Early Event Detector (MMED) [12], and Dynamic Bag-of-Words [27]. The last two methods were specifically proposed to address early recognition problems.

The proposed method and the ones being compared to have different recognition philosophies. We therefore implemented and optimized each method based on its preferred representation of the human skeleton sequence, i.e., a feature representation that is commonly used and well suited for the method being evaluated. The evaluation of all methods is of course carried out using identical data. The proposed method is the combination of two integrated LSTMs; the input to the LSTMS is the sequence of vectors of 3D coordinates of the human body joints. The input features for SVM, SOSVM, and MMED were based on sparse coding [21, 40] and temporal pyramid pooling [23]. This type of features has been shown to work well for the recognition task with max-margin classifiers [23].

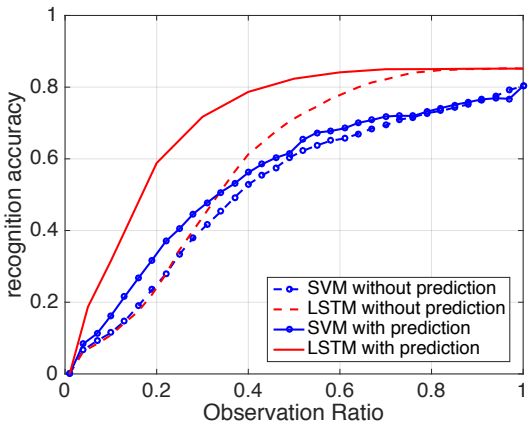
To use sparse coding features, we first learned a dictionary for encoding skeleton data. Given a training set $\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_N]$, where each $\mathbf{s}_i$ represents the skeleton vector for one pose, sparse dictionary learning learns a dictionary by optimizing:

$$\min_{\mathbf{D},\{\boldsymbol{\alpha}_i\}} \sum_{i=1}^{N} \left(||\mathbf{s}_i - \mathbf{D}\boldsymbol{\alpha}_i||_2^2 + \lambda |\boldsymbol{\alpha}_i|_1 \right), \qquad (1)$$

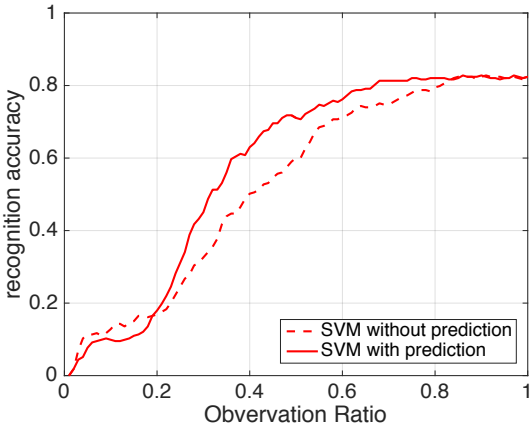
where the matrix $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_M]$ was the dictionary with M atoms and $\boldsymbol{\alpha}_i$ was a sparse vector of coefficients for encoding the training pose $\mathbf{s}_i$ as a sparse linear combination of atoms in the dictionary. Once the dictionary had been learned, a feature vector for a sequence of human poses was computed as follows: 1) used the learned dictionary to encode individual poses; 2) divided the sequence into two halves, and divided each half into two halves again (temporal pyramid with two layers); 3) within each segment, used max pooling to compute a the feature vector for the segment, and subsequently concatenated all feature vectors to represent the entire sequence.

D. Early recognition

We first studied the benefits of body movement prediction for early recognition of human actions. We analyzed the performance of recognition methods with and without body movement prediction. We also compared with several early recognition methods [12, 27].



(a) Results on Montalbano dataset

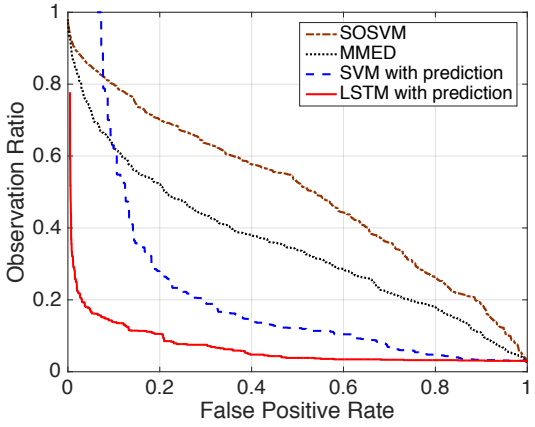


(b) Results on MSR Action 3D dataset

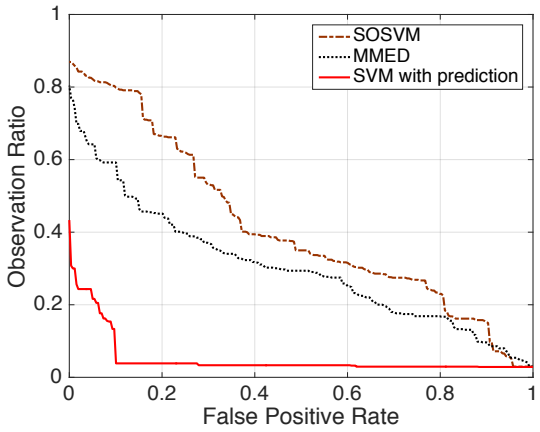
Fig. 3: **Benefits of movement prediction for early recognition.** This figure shows the performance of two methods (SVM and LSTM) with and without movement prediction. The horizontal axis shows the proportion of an action that has been observed. The vertical axis shows the recognition accuracy, a higher value means a better performance. The horizontal axis shows the observational ratio (i.e., the proportion of the action that has been observed at the time of decision). Movement prediction provides benefits to both recognition methods on both datasets.

We considered the ability for early human action recognition, i.e., recognizing an action when only the beginning portion of the action has been observed. Given a partial sequence, we jointly used the prediction network to complete the sequence and used the early recognition network for classification, as illustrated in Fig. 2. The prediction network was trained on the validation set, and we use a 3-layer LSTM with the memory size of 300 (dimension of the memory vectors).

Figure 3 compares the early recognition performance of two approaches, with and without using movement prediction. For each approach, two types of recognition classifiers were considered: (1) SVM with sparse coding and temporal pyramid pooling and (2) the LSTM recognition network. The LSTM recognition network has five hidden layers with RNN size of 300, and the network was trained on the train



(a) Results on Montalbano dataset



(b) Results on MSR Action 3D dataset

Fig. 4: **Comparison of several methods for early event detection.** These figures show the AMOC curves for binary detection task. MMED is a method that is proposed for early event detection. Although it works better than SOSVM, it performs worse than LSTM and SVM methods that use the predicted body movement.

dataset. For MSR Action 3D dataset, there were only 284 sequences available for training, so we did not use an RNN recognition system to avoid overfitting. The results reported in Figure 3b are based on SVM with sparse dictionary learning and temporal pyramid pooling instead. As can be seen, using movement prediction significantly improves early recognition performance. This applies to both SVM and LSTM classifiers.

Figure 4 compares SVM and LSTM with movement prediction with two other methods for early recognition that do not use movement prediction: Structured-Output SVM [32] and Max-Margin Early Event Detector (MMED) [12]. Since MMED is a binary event detector, we adapted our multi-class recognition method to the binary event detection task. Following [12], we use the Activity Monitoring Operating Characteristic (AMOC) curve to evaluate the timeliness of detection. AMOC curve shows the relationship between False Positive Rate (FPR) and the observational ratio (i.e., the proportion of the action that has been observed at the time

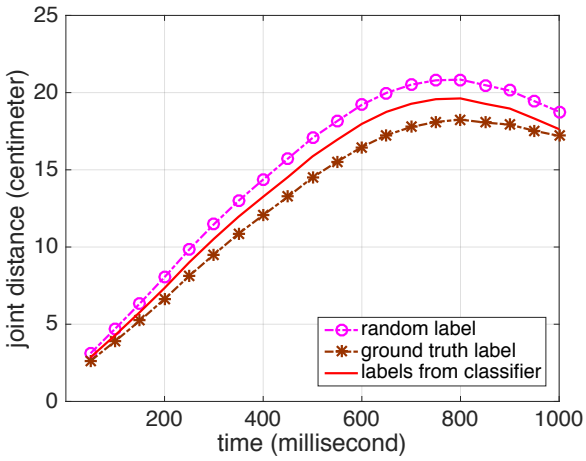


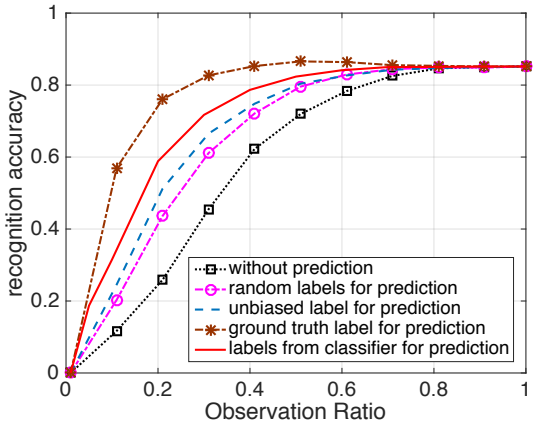
Fig. 5: **Motion prediction performance using different class probability vectors.** This figure shows the average prediction error for the joints on the arms (distance between the actual location and the predicted location). The distance measure is in centimeter, assuming the average person height is 1.7m. The prediction horizon is 1000ms. The prediction network PredLSTM can generate a sequence of body movement conditioned on a class probability vector. Knowing the true label is very useful; it yields the lowest prediction error. Using the predicted class probability is better than the uniform class probability.

of decision). By adjusting the detection threshold, one can detect the action sooner at the cost of higher FPR and vice versa. For a complete picture, we vary the detection threshold and plot the curve of observational ratio versus FPR. Figure 4 shows the comparison between several methods: LSTM with movement prediction, SVM with movement prediction, Structured-Output SVM, and MMED. For each action class, we consider the binary detection task and there is a set of corresponding AMOC curves. Figure 4 shows the AMOC curves for two representative classes of the two datasets. As can be seen, the proposed recognition approach (either LSTM or SVM) that uses the predicted movement can detect the actions faster than SOSVM and MMED at the same FPR.

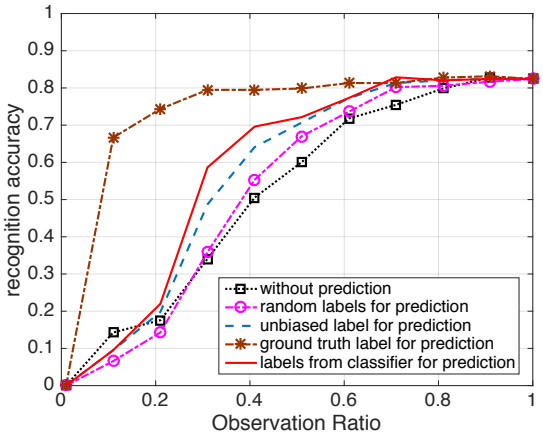
We also implemented the Dynamic Bag-of-Words method [27]. We found that Dynamic Bag-of-Words did not work well for skeleton data. Even for offline recognition (not early recognition), the classification accuracy was low. We experimented with several parameter settings of the method, but the best accuracy was only 24.9%. This is perhaps due to the sparsity of the Bag-of-Words feature histograms constructed based on a pose vocabulary. The Dynamic Bag-of-Words method has been shown to work well for the UT-Interaction dataset [28] where dense spatio-temporal interest points were used. Dense spatial-temporal interest points, however, are not available to a sequence of human skeletons.

E. Prediction analysis

In this experiment, we analyzed the impact of the class probability vector (denoted as $\mathbf{p}_t$ in Fig. 2) to the prediction network and its influence on early recognition performance.



(a) Montalbano - early recognition



(b) MSR Action - early recognition

Fig. 6: **The impact of using different class probability vectors on early recognition performance.** (a): comparison in Montalbano dataset; LSTM is used as the recognition method. (b): comparison in MSR Action 3D dataset; SVM is used as the recognition method. The key difference is how the unseen sequence is synthesized. The prediction network PredLSTM can generate a sequence of body movement conditioned on a class probability vector. We compare the four settings for the class probability vector: 1) use a random class label; 2) use uniform class label; 3) use the ground truth label; and 4) use the class probability vector that is produced by the recognition network. Knowing the true label is very useful; it yields the highest recognition accuracy. Using the predicted class probability vector is better than the uniform or random class probability vector.

Recall that our prediction network can generate a sequence of body movement conditioned on a class probability vector. We compared the following settings for the class probability vector: 1) use the binary indicator vector for the true class label as the input to our prediction network—this is an ideal case where the class label is known; 2) use the class label vector that has uniform weights (no bias to any class); 3) use a random class label that is different from the true label; 4) use the class probability vector that is produced by the recognition network (proposed).

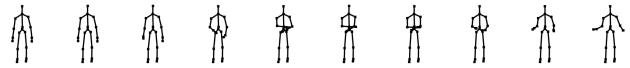


Fig. 7: **The initial 20 frames** used to generate sequences in Figure 8. We show every two frames.

Figure 5 shows the average prediction error for the joints on the arms. Using the ground truth class label leads to the lowest error. Using the probability vector produced by the recognition network also performs relatively well. The average distance for using ground truth label, labels from classifier output, and random label are 13.43 cm, 14.50 cm, and 15.54 cm respectively.

Figure 6a compares the early recognition performance of the LSTM recognition network when pairing with different predicted movement sequences. Interestingly, not using a predicted sequence has the worst performance. This confirms the benefits of movement prediction for early recognition. There is a big difference between using the ground truth label and other methods. This indicates the importance of the class probability vector for the prediction network. Figure 6b show the same analysis on the MSR Action 3D dataset, and similar conclusions can be drawn.

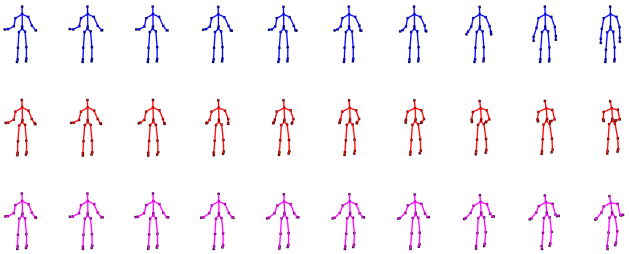


Fig. 8: **Ground-truth (1^{st} row) and predicted sequences (2^{nd} and 3^{rd} rows)** using the sequence shown in Figure 7 as the initial states. The second row and third row are predicted sequences from frame 21 to 40 after seeing the sequence in Figure 7. The second row is predicted using uniform class probability vector. The third row is predicted by the proposed joint prediction and recognition network. We show a pose at every two frames.

F. Qualitative analysis of prediction

We show our prediction qualitatively. Given the partial sequence shown in Figure 7, we used different class label vectors to predict the rest of the sequence. As shown in Figure 8, the first row is the true pose sequence from frame 21 to 40 (every second frame is shown). The second row and third row are predicted sequences after seeing the sequence in Figure 7. The second row was predicted using uniform class probability vector. The third row was generated by the proposed joint prediction and recognition network.

V. Conclusions and Discussions

We have demonstrated that movement prediction and early recognition of human action provide mutual benefits to each other. These two tasks have traditionally been addressed separately, and it is difficult to combine different methods

that were designed for individual tasks because they do not expect and accept inputs from each other. In this paper, we have considered the case where 3D skeleton information is available and have presented a framework for joint movement prediction and early recognition. Our framework integrates two Long Short-Term Memory recurrent neural networks, which are specifically designed to incorporate the mutual benefits of each other. Experiments on two human action datasets showed that the integrated system outperformed the individual subsystems. The integrated system also outperformed several state-of-the-art methods for early recognition.

One limitation of our current work is that we only have the empirical demonstration for skeleton-based representation. However, we believe the mutual benefits of prediction and early recognition exist beyond skeleton-based representation. One possible direction for future work is to extend the current framework for analyzing RGB sequences, using the recent human pose estimation methods [4, 11, 30, 39].

REFERENCES

[1] M. S. Aliakbarian, F. S. Saleh, M. Salzmann, B. Fernando, L. Petersson, and L. Andersson. Encouraging lstms to anticipate actions very early. In *Proc. ICCV*, 2017.

[2] M. Brand and A. Hertzmann. Style machines. In *Proc. ACM SIGGRAPH*, 2000.

[3] S. Cao and R. Nevatia. Forecasting human pose and motion with multibody dynamic model. In *Proc. WACV*, 2015.

[4] Z. Cao, S.-E. Wei, T. Simon, and Y. Sheikh. Realtime multiperson pose estimation. In *Proc. ECCV*, 2016.

[5] A. D. Dragan and S. S. Srinivasa. *Formalizing assistive teleoperation*. MIT Press, July, 2012.

[6] Y. Du, W. Wang, and L. Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *Proc. CVPR*, 2015.

[7] C. Ellis, S. Z. Masood, M. F. Tappen, J. J. Laviola Jr, and R. Sukthankar. Exploring the trade-off between accuracy and observational latency in action recognition. *IJCV*, pages 420–436, 2013.

[8] S. Escalera, X. Baró, J. Gonzalez, M. A. Bautista, M. Madadi, M. Reyes, V. Ponce-López, H. J. Escalante, J. Shotton, and I. Guyon. Chalearn looking at people challenge 2014: Dataset and results. In *Proc. ECCV Workshops*, 2014.

[9] P. Felsen, P. Agrawal, and J. Malik. What will happen next? forecasting player moves in sports videos. In *Proc. ICCV*, 2017.

[10] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik. Recurrent network models for human dynamics. In *Proc. ICCV*, 2015.

[11] G. Gkioxari, A. Toshev, and N. Jaitly. Chained predictions using convolutional neural networks. In *Proc. ECCV*, 2016.

[12] M. Hoai and F. De la Torre. Max-margin early event detectors. In *Proc. CVPR*, 2012.

[13] M. Hoai and F. De la Torre. Max-margin early event detectors. *IJCV*, 107(2):191–202, 2014.

[14] M. Hoai and A. Zisserman. Improving human action recognition using score distribution and ranking. In *Proc. ACCV*, 2014.

[15] M. Hoai and A. Zisserman. Talking heads: Detecting humans and recognizing their interactions. In *Proc. CVPR*, 2014.

[16] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[17] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena. Structural-rnn: Deep learning on spatio-temporal graphs. In *Proc. CVPR*, 2016.

[18] K. M. Kitani, B. Ziebart, D. Bagnell, and M. Hebert. Activity forecasting. In *Proc. ECCV*, 2012.

[19] H. S. Koppula and A. Saxena. Anticipating human activities for reactive robotic response. In *Proc. IEEE/RSJ Conf. on Intelligent Robots and Systems*, 2013.

[20] M. Kuderer, H. Kretzschmar, C. Sprunk, and W. Burgard. Feature-based prediction of trajectories for socially compliant navigation. In *RSS*, 2012.

[21] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *NIPS*, 2006.

[22] W. Li, Z. Zhang, and Z. Liu. Action recognition based on a bag of 3d points. In *CVPR Workshop*, 2010.

[23] J. Luo, W. Wang, and H. Qi. Group sparsity and geometry constrained dictionary learning for action recognition from depth maps. In *Proc. ICCV*, 2013.

[24] V. Pavlovic, J. M. Rehg, and J. MacCormick. Learning switching linear models of human motion. In *NIPS*, 2000.

[25] S. Qi, S. Huang, P. Wei, and S.-C. Zhu. Predicting human activities using stochastic grammar. *Proc. ICCV*, 2017.

[26] M. Raptis and L. Sigal. Poselet key-framing: A model for human activity recognition. In *Proc. CVPR*, 2013.

[27] M. Ryoo. Human activity prediction: Early recognition of ongoing activities from streaming videos. In *Proc. ICCV*, 2011.

[28] M. Ryoo and J. K. Aggarwal. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In *Proc. ICCV*, 2009.

[29] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014.

[30] J. Song, L. Wang, L. Van Gool, and O. Hilliges. Thin-slicing network: A deep structured model for pose estimation in videos. 2017.

[31] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proc. ICCV*, 2015.

[32] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *JMLR*, 6:1453–1484, 2005.

[33] C. Vondrick, H. Pirsiavash, and A. Torralba. Anticipating the future by watching unlabeled video. In *Proc. CVPR*, 2016.

[34] J. Walker, K. Marino, A. Gupta, and M. Hebert. The pose knows: Video forecasting by generating pose futures. In *Proc. ICCV*, 2017.

[35] J. M. Wang, D. J. Fleet, and A. Hertzmann. Gaussian process dynamical models for human motion. *IEEE PAMI*, 2008.

[36] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: towards good practices for deep action recognition. In *Proc. ECCV*, 2016.

[37] Y. Wang and M. Hoai. Improving human action recognition by non-action classification. In *Proc. CVPR*, 2016.

[38] Z. Wang, M. P. Deisenroth, H. B. Amor, D. Vogt, B. Schölkopf, and J. Peters. Probabilistic modeling of human movements for intention inference. In *RSS*, 2012.

[39] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *Proc. CVPR*, 2016.

[40] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *Proc. CVPR*, 2009.

[41] M. Zanfir, M. Leordeanu, and C. Sminchisescu. The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection. In *Proc. ICCV*, 2013.

[42] K.-H. Zeng, W. B. Shen, D.-A. Huang, M. Sun, and J. C. Niebles. Visual forecasting by imitating dynamics in natural sequences. 2017.

[43] B. D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. A. Bagnell, M. Hebert, A. K. Dey, and S. Srinivasa. Planning-based prediction for pedestrians. In *Proc. IEEE/RSJ Conf. on Intelligent Robots and Systems*, 2009.