3D Action Recognition by Learning Sequences of Poses

Srinivas S S Kruthiventi*
Video Analytics Lab
Supercomputer Education and Research Centre
Indian Institute of Science, Bangalore, India
kssaisrinivas@gmail.com

R. Venkatesh Babu
Video Analytics Lab
Supercomputer Education and Research Centre
Indian Institute of Science, Bangalore, India
venky@serc.iisc.ernet.in

ABSTRACT

Action recognition from 3D depth maps/skeleton has been an active research area in computer vision in the recent past. In this paper, we propose a method for action recognition by learning the sequence of various poses involved while performing an action. Each pose in the sequence is represented as a set of 3D edge vectors connecting important joint points in the skeleton and 3D trajectory vectors connecting the joint point locations in the previous frame with that in the current frame. The training samples of each action class, represented as sequences of poses, are time normalized using Dynamic Time Warping(DTW) and average pooled to construct the model sequence representing the action. Action recognition on a test sequence is achieved by finding its nearest model sequence in terms of a proposed distance measure. The effectiveness of the proposed algorithm is evaluated on two challenging datasets: Berkeley Multimodal Human Action Database and MSR Action 3D dataset.

Keywords

Action recognition, Skeleton joints, Dynamic Time Warping(DTW)

1. INTRODUCTION

Recognizing human actions has been an active research problem in computer vision since its inception. The importance of action recognition is evident from its applications in diverse fields. In visual surveillance, action recognition can help in building intelligent surveillance systems and in detecting anomalous activities. In Human-Computer Interaction (HCI), action recognition can enable smart environments and enhance user experience.

Despite being a well-researched area, algorithms for action recognition perform poorly when compared to humans on real-world datasets like HMDB[3][2]. The task of action recognition on real-world videos can be challenging due to many reasons like occlusion, view point changes, individual variances, background clutter, illumination changes etc. To circumvent some of these video-related problems, researchers have recently been working on datasets having annotated joint points[4][7]. Using joint-annotated datasets decouples the problem of action recognition from pose estimation. The essential high-level information required for recognition is readily available in the form of joint point locations. This approach gives an emphasis towards finding an effective solution to the problems of view point changes, individual variations and similarity between different actions overcoming the problems of background clutter and illumination changes. With the advent of low cost depth sensors like Microsoft Kinect and its Motion Capture system which can provide joint point locations in real-time, solutions of this kind become more relevant.

In the present work, we have proposed an algorithm for action recognition using 3D skeleton data. This 3D skeleton data can be either extracted from depth map sequences using real-time skeleton tracking algorithm[9] or directly obtained using Motion Cature(MoCap) systems. In MoCap systems, LED markers are attached to subject's body at various joint locations. These markers are tracked continuously to obtain their 3D coordinates as the subject performs various actions. We extract features from each frame of this skeleton data to represent pose effectively. A model sequence of poses is learnt for each action class by average pooling over the time-warped training data of that class. The proposed approach performs action recognition by finding a distance measure for a given test action sequence with the model sequences learnt for each of the action classes. We have evaluated our algorithm on the publicly available Berkeley Multimodal Human Action Database and MSR Action 3D dataset.

The rest of the paper is organized as follows. Section 2 presents a short review of recent work in action recognition using 3D joint point locations. Section 3 describes the proposed algorithm of learning sequence of poses for action recognition. Experimental results are presented in Section 4. We conclude with a summary of the proposed method in Section 5.

2. RELATED WORK

During recent times, researchers have started working on action recognition databases having annotated 3D skeletal joint points. In [7], Ofli *et al.* proposed Berkely Multimodal Human Database (MHAD) which consists of human actions captured using multiple modalities like Motion Capture, Kinect, multi-stereo cameras etc. In the same work,

*Corresponding author

ICVGIP '14, December 14-18, 2014, Bangalore, India

http://dx.doi.org/10.1145/2683483.2683506.

the authors proposed a Bag-of-Words(BOW) model on joint angles obtained from MoCap data for action recognition. They divided each action sequence into temporal windows and a concatenation of variances of skeleton joints in each window was taken as a local feature descriptor. They also demonstrated how fusion of data from all the modalities can help in achieving a higher recognition rate. In [6], the same authors proposed an algorithm for recognition using histograms of most informative joints i.e., joints with high variances.

Later in [11], Vantigodi *et al.* proposed a method which uses time weighted variances along with temporal variances of joints for action recognition.Here, the time weighted variances help in distinguishing between mirror actions like sit-down and stand-up. These variances were used to obtain a feature representation for each action sequence which are classified using Support Vector Machines(SVM). In [12], the same authors used Meta-Cognitive Radial Basis Function Network (McRBFN) and its Projection Based Learning (PBL) algorithms for action recognition. A 129 dimensional vector, constructed by considering the 3D angles made by each of the joints with a fixed point on the skeleton, was used as a skeleton representation for each frame. Each action sequence was represented using a bag-of-words model of these 129 dimensional vectors. An additional feature of temporal variance of joints was considered to represent the action sequence.

In [4], Li *et al.* proposed MSR Action 3D dataset of 20 actions captured using a depth sensor like Kinect. In the same work, the authors proposed an action graph to model the dynamics of actions for recognition. Here nodes of the graph correspond to a bag of 3D points sampled from depth maps to characterize a set of salient postures. In [14], Wang *et al.* proposed a method of learning an actionlet ensemble to represent each action. These actionlets were built based on 3D joint locations and local occupancy patterns which capture the interaction between the human subject and environmental objects.

3. ACTION RECOGNITION BY LEARNING SEQUENCE OF POSES

Any human action is a time sequence of specific set of body poses. The sequence of body poses differs for different actions, but remains largely the same for different people performing the same action. For example, the action of *waving the hand* involves - raising the arm, bending it at a specific angle at the elbow, moving the forearm to and fro in a specific direction. So the action associated with a given test sequence of poses can be recognized by finding its distance/dissimilarity with already labelled sequence of poses from training data.

3.1 Pose Representation and Sequence Warping

In video/MoCap systems, each frame captures the body motion at a time instant and thus represents a single pose. Here in this work, we have formulated a similarity measure for poses. Using this measure we have proposed a method to estimate distance/dissimilarity between two pose sequences. Comparing two sequences of poses involves two steps:

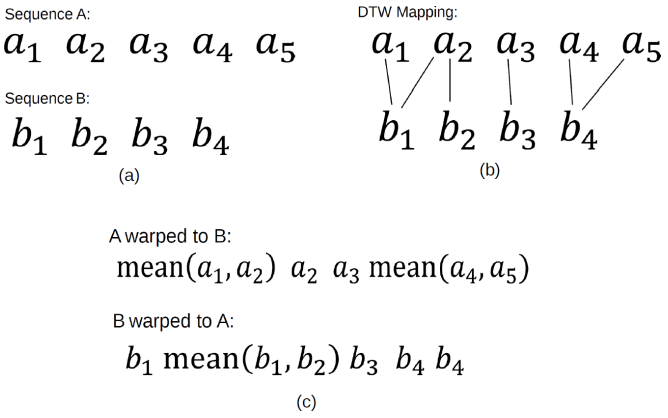

Figure 1: (a) Sequences A and B (b) Dynamic Time Warping of A and B (c) Sequence A warped to B and sequence B warped to A

- measuring pose similarity
- finding frame to frame mapping between the two sequences

(i) Pose Representation and Distance: To represent a skeleton with n joint points, we have considered a set of n trajectory vectors and n edge vectors:

Trajectory Vectors: They correspond to the trajectories of n joint points over time. These vectors are constructed by connecting each joint point location in the previous time instant to that in the current time instant.

Edge Vectors: An initial set of edges are formed by connecting each joint point with all other joint points in the skeleton. Like the trajectory vectors, these edges are also 3 dimensional vectors in the real world space(x,y,z coordinates). Each edge is assigned a score equal to the mean of its temporal variance across all the training samples from all actions. Out of all the edges incident on each joint point, we pick the edge with maximum score (see Figure 4). Doing this for all the n joint points, we get a total of n edge vectors.

These n edges may not be unique i.e., an edge may be repeated twice. For e.g., in MSR3D dataset, the highest scoring edge corresponding to the joint point *R. Palm* is the one connecting *R. Palm* and *R. Ankle*. The highest scoring edge corresponding to *R. Ankle* also happens to be the same one. So this edge appears twice in the set of n edges i.e., a relative weight of 2 is given while emphasizing the more active joints.

With this representation, dissimilarity between two poses $P : \{\{e_P^i\}_{i=1}^n, \{t_P^i\}_{i=1}^n\}$, $Q : \{\{e_Q^i\}_{i=1}^n, \{t_Q^i\}_{i=1}^n\}$ can be computed as follows:

$$distance(P,Q) = \sum_{i=1}^{n} \parallel e_P^i - e_Q^i \parallel_2 + \alpha \sum_{i=1}^{n} \parallel t_P^i - t_Q^i \parallel_2 \quad (1)$$

where e_P^i, e_Q^i are edge vectors and t_P^i, t_Q^i are trajectory vectors of poses P and Q respectively. α is the relative weight given to trajectory vectors.

Since a trajectory vector connects a joint point in two successive frames, its magnitude will generally be less com-

Algorithm 1 : ***Distance between sequences of poses***

Require: 2 pose sequences: U = $\{u_i\}_{i=1}^{N_u}$, V = $\{v_j\}_{j=1}^{N_v}$
%% N_u, N_v : no. of frames in sequences U and V
Ensure: distance(U,V)

%% Construct the cost matrix of size $N_u \times N_v$
for $i = 1 \rightarrow N_u$ **do**
  for $j = 1 \rightarrow N_v$ **do**
    $C(i,j) = \sum_{l=1}^{n} \parallel e^l_{u_i} - e^l_{v_j} \parallel_2 + \alpha \sum_{l=1}^{n} \parallel t^l_{u_i} - t^l_{v_j} \parallel_2$
  end for
end for

%% Perform Dynamic Time Warping(DTW) of U & V with the cost matrix C
[FrameMapping, CumulativeCost] = DTW(C);
distance(U,V) = CumulativeCost;

Algorithm 2 : ***Learning Model Sequence of Poses***

Require: N Pose Sequences for training: $\{S^i\}_{i=1}^{N}$
Ensure: Model Pose Sequence: S_{model}

%% Assume the training sequence with least no. of frames as initial model
$S_{model} = S^j \; s.t. \; j \in [1 \; N] \; \& |S_j| \leq |S_k| \forall \; k \in [1 \; N]$

while !$convergence(S_{model})$ **do**
  for $i = 1 \rightarrow N$ **do**
    %% Time align the sequence to model using DTW as shown in Figure 1
    $S^i_{warp} = warp(S^i, S_{model})$
  end for

  %% Update the model to be mean of all warped sequences
  $S_{model} = mean(\{S^i_{warp}\}_{i=1}^{N})$
end while

pared to that of the edge vectors which connect different joint points in the same frame. So the trajectory vectors are scaled up by a factor of 10, *i.e.,* $\alpha = 10$, to give them equal importance in the distance computation in Eq.1. The performance of our algorithm is stable with respect to small changes in this scaling factor.

(ii) Frame Level Correspondence: The importance of this task can be understood from the fact that the same action can be performed at different rates and that this rate need not be constant throughout the sequence. So before comparing any two sequences of poses they need to be appropriately time normalized. This corresponds to finding a frame level mapping between the two sequences. This task is often popularly referred to as Dynamic Time Warping (DTW) in the speech community and is used for matching two speech sequences[8].

Given two sequences of features $A = \{a_i\}_{i=1}^{M}, B = \{b_j\}_{j=1}^{N}$, their cost matrix of order MxN can be defined as $C(i,j) = distance(a_i, b_j)$. With this cost matrix, feature mapping and dissimilarity/distance of sequences can be found out using DTW [10]. This is illustrated in Figure 1. The process of time warping results in a mapping with minimum cumulative cost over the mapped features while ensuring that the mapping is both continuous and monotonic. Hence DTW is an efficient way to find frame level mapping between the two sequences of poses. Here the cost matrix C is constructed using the pose distance measure discussed in Eq.1.

With this formulation, a distance measure between two sequences of poses can be computed as shown in the Algorithm 1.

3.2 Learning Sequence of Poses

A simple way to use the formulated distance measure of sequence of poses for action recognition would be by using an SVM classifier. The proposed distance measure can be used to construct a kernel matrix of training and test sequences of all the action classes which can be further used to train a multi-class Support Vector Machine (SVM). This approach, though works reasonably well, has the drawback of a high computational cost. Since the distance of a given test sequence needs to be computed with each of the training sequences to construct the kernel, the computation time needed for recognition of a test sequence increases linearly with increase in the amount of training data available.

To overcome this computational demand, we have created a model sequence of poses to represent each action class by learning from its training data. The optimization problem of learning sequence of poses can be formulated as

$$S_{model} = \arg\min_{S} \sum_{i=1}^{N} distance(S, S^i) \tag{2}$$

where $\{S^i\}_{i=1}^{N}$ are the N training sequences for an action class.

To obtain S_{model}, we start by choosing the training sequence of least length as our initial model. All the training sequences are warped to this model as illustrated in Figure1. These warped sequences are then average pooled across each frame to obtain the new model. This process is repeated until the model converges. This is explained in detail in Algorithm 2.

In [5], where Dynamic Time Warping was used to learn Motion Templates(MT), each iteration would require N^2 operations of DTW whereas the proposed method would only need to perform N operations of DTW. The proposed method also usually converges in less than 15 iterations.

With the model sequence learnt for each of the action classes, a test sequence is labeled with the action to whose model it is least distant.

It is well known that the mean serves as a good model for data corrupted by gaussian noise. Since our model for each action is constructed by mean pooling over the time-warped training samples, it is robust to intra-class variance and can handle noise present in the acquisition/estimation of joint point coordinates of the skeleton to a certain extent.

4. EXPERIMENTS AND RESULTS

In this section, experimental evaluation of the proposed method is discussed. We have evaluated our algorithm on two different datasets: Berkeley MHAD database[7] and MSR Action 3D dataset[4]. We present the results of our proposed method along with that of other state-of-the-art algorithms.

Figure 2: Different Actions in MHAD database - jump, jumpingjacks, bend, punch, wave - 2 hands, wave, clap, throw, sit down&stand up, sit down, stand up

4.1 Berkeley MHAD database

Berkeley Multimodal Human Action Database(MHAD) consists of 11 different actions - *jump, jumpingjacks, bend, punch, wave - 2 hands, wave, clap, throw, sit down&stand up, sit down, stand up.* These actions are performed by 12 different subjects (7 male and 5 female). Each subject repeats all the 11 actions 5 times giving a total of 660 action sequences. These actions are illustrated in Figure 2. All subjects are given instructions only on what action to perform but not on the specific details concerning how the action should be executed (i.e., performance style or speed). Thus the subjects have incorporated their individual styles in performing some of the actions (e.g., punching, throwing).

Table 1: Results of various algorithms on MHAD database

Algorithm	Recognition Accuracy
Kurillo *et al.*[7]	79.93
Babu *et al.*[11]	96.06
Vantigodi *et al.*[12]	97.58
Proposed Method	**98.33**

The motion of the subject performing the action is captured by 5 different modalities - PhaseSpace Motion Capture (MoCap), multi-stereo cameras, Microsoft Kinect, accelerometers and microphones. We have used Motion Capture data for all of our experiments. The Motion Capture data is acquired by tracking 43 LED markers attached to various parts of the subject's body using the optical motion capture system Impulse(PhaseSpace Inc., CA).

In leave-one-out approach, 600 action sequences from 11 subjects are considered for training and the left out subject's 60 action sequences are used for testing. The performance is

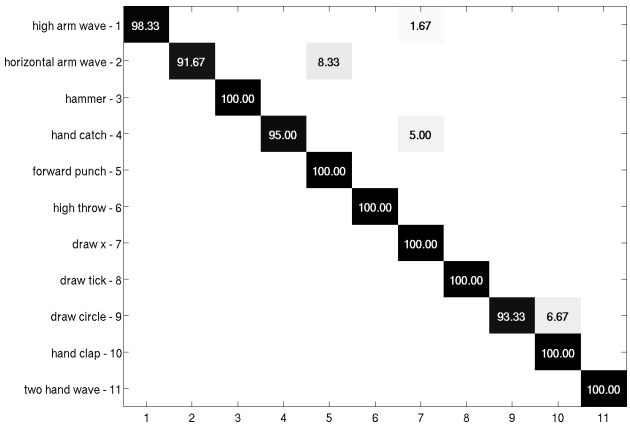

Figure 3: Results obtained by proposed algorithm on MHAD database

then computed by averaging the results of each of the 12 test subjects. We adopted this approach to facilitate comparison of our algorithm's performance with other state of the art methods. In [7], the authors have achieved a recognition accuracy of 79.93% using MoCap modality alone and 93.8% using Kinect and MoCap modalities. Vantigodi *et al.*[11], have obtained a recognition accuracy of 96.06% using temporal variances of joint points. The same authors[12] also reported a recognition accuracy of 97.58% using Meta-cognitive RBF Network Classifiers. We have achieved a recognition accuracy better than other state of the art methods using the proposed algorithm. Our method of learning sequence of poses gives a recognition accuracy of 98.33% using the MoCap modality alone. These results are presented in Table 1.

4.2 MSR Action 3D dataset

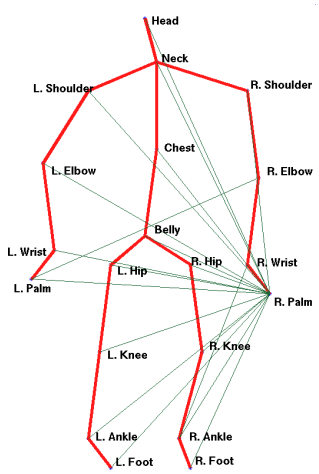

Figure 4: Edge vectors used for pose representation in MSR 3D dataset

The MSR Action 3D dataset consists of following 20 actions: *high arm wave, horizontal arm wave, hammer, hand catch, forward punch, high throw, draw x, draw tick, draw circle, hand clap, two hand wave, side-boxing, bend, forward kick, side kick, jogging, tennis swing, tennis serve, golf swing, pickup&throw.* These 20 actions are performed by 10 subjects. Each subject repeats each action 2-3 times yielding a total 567 depth map action sequences. The depth sequences are recorded using a depth sensor similar to the Kinect device, sampled at 15Hz. From these depth sequences, skeleton points can be extracted in real-time using the algorithm proposed in [9]. The skeleton data obtained from these depth maps is publicly available along with the original dataset. This database is more challenging as depth acquisition is noisy in certain scenarios, such as when the subject moves close to a sofa or sits on it. We also encounter the problem of occlusion when the subject goes behind a sofa.

In MSR Action 3D dataset, the extracted skeleton from depth maps contains 20 joint points. For this dataset, the 20 edge vectors connecting active joints are shown in green over the red human skeleton in Figure 4. Since this dataset majorly consists of actions involving right hand, most of the edge vectors can be observed to have the right wrist or the right palm as one of their end points. It can also be observed that, using the proposed pose representation, the whole body pose is captured while emphasizing the active joints.

We have tested the proposed algorithm on this dataset using cross-subject evaluation. As in [4] and various other works [15][13][14], we have considered the subjects 1, 3, 5, 7, 9 for training and the rest for testing the performance of the algorithm. Further, the 20 actions are grouped into 3 action sets - AS1, AS2, AS3, where each action set contains 8 actions. The first two sets group actions with similar movements together whereas the third set groups complex actions[4]. We have performed recognition using the proposed algorithm independently on each of these action sets and the results are presented in Table 2 and Figure 5.

In [14], a more challenging experimental setup of performing recognition on all the 20 actions is considered. However,

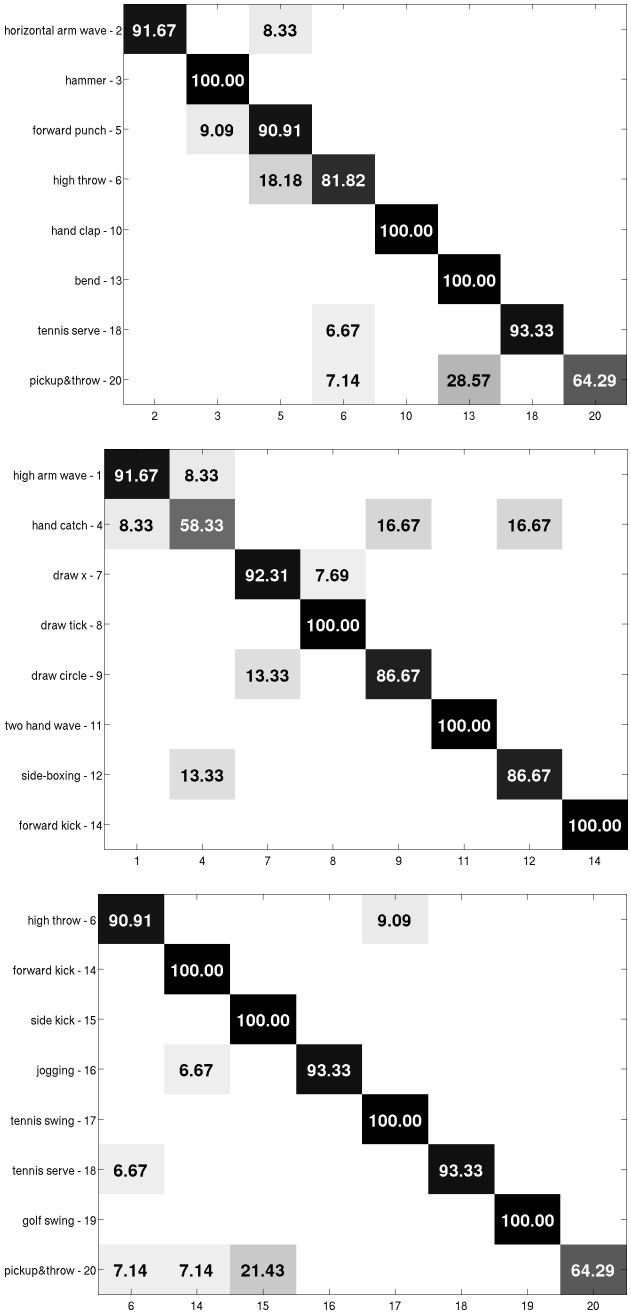

Figure 5: Confusion matrices of the proposed method for MSR 3D AS1, AS2 and AS3

here only 557 of the total 567 sequences are used by discarding 10 sequences which have missing or completely wrong skeletons. The proposed algorithm is also evaluated on this experimental setup and the results are presented in Table 3 and Figure 6.

The proposed method is seen to perform poorly for the actions - *hammer, hand catch and pickup&throw.* It is observed that in case of the *hammer* action, all but one of the training subjects swing the hammer once in each sequence thereby biasing the learnt model. However, a significant number of test samples swing the hammer twice in each se-

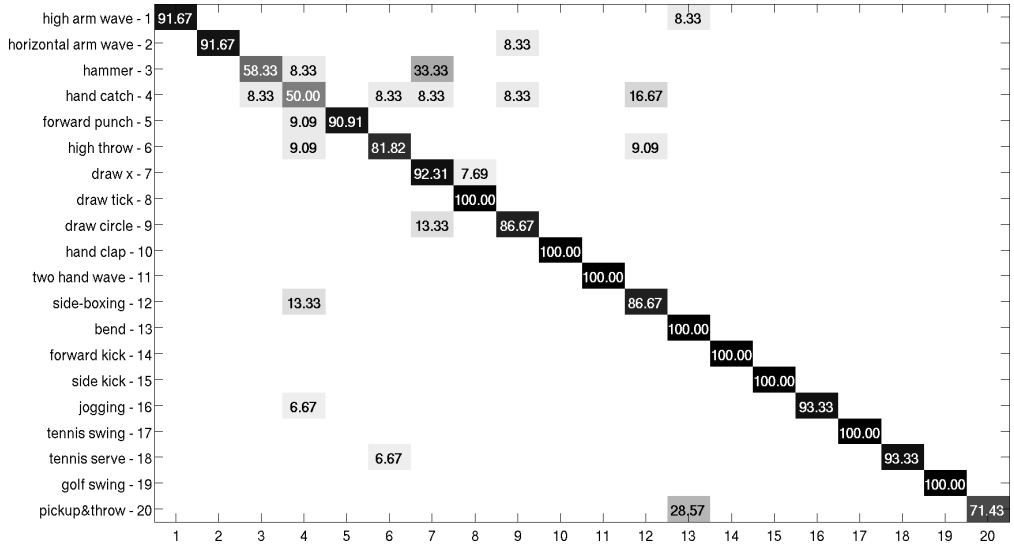

Figure 6: Confusion Matrix of the proposed method on MSR Action 3D dataset

Table 2: Recognition accuracies on the 3 action sets of MSR Action 3D dataset

Algorithm	AS1	AS2	AS3	Average
Histogram of Joints[15]	87.98	85.48	63.46	78.97
Eigen Joints[16]	74.5	76.1	96.4	82.3
Covariance of Joints[1]	88.04	89.29	94.29	90.53
Lie Group[13]	95.29	83.87	98.22	92.46
Proposed Method	**90.57**	**89.38**	**92.86**	**90.93**

Table 3: Recognition accuracies on all the 20 actions of MSR Action 3D dataset

Algorithm	Recognition Accuracy
Mining Actionlets [14]	88.20
Lie Group[13]	89.48
Proposed Approach	**90.11**

quence leading to confusion with the *draw x* action. In case of the *pickup&throw*, the test sequences are observed to be so noisy that entire skeleton collapses to a single point after certain frames, causing confusion with the *bend* action.

5. CONCLUSION

In this work, we have proposed an algorithm for action recognition using 3D skeleton data. Our algorithm represents an action as a sequence of poses. Each pose is represented by vectors connecting active joint points of human skeleton and joint point trajectories. For each action class, we learn a model sequence of poses by using techniques of Dynamic Time Warping (DTW) and mean pooling. Action recognition is performed by computing the proposed distance measure of the test sequence with each of the model sequences. The proposed method is shown to achieve good recognition accuracies on the widely used MSR Action 3D dataset and the Berkeley MHAD database. This can be attributed to the robustness of our novel features and the technique of construction of the model.

6. REFERENCES

[1] M. E. Hussein, M. Torki, M. A. Gowayyed, and M. El-Saban. Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, pages 2466–2472. AAAI Press, 2013.

[2] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. Black. Towards understanding action recognition. In *IEEE International Conference on Computer Vision,*, pages 3192–3199, Dec 2013.

[3] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *International Conference on Computer Vision*, 2011.

[4] W. Li, Z. Zhang, and Z. Liu. Action recognition based on a bag of 3D points. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 9–14, June 2010.

[5] M. Müller and T. Röder. Motion templates for automatic classification and retrieval of motion capture data. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '06, pages 137–146, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association.

[6] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy. Sequence of the most informative joints (SMIJ): A new representation for human skeletal action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops,*, pages 8–13, June 2012.

[7] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy. Berkeley mhad: A comprehensive

multimodal human action database. In *IEEE Workshop on Applications of Computer Vision*, pages 53–60, 2013.

[8] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing,*, 26(1):43–49, Feb 1978.

[9] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 1297–1304, Washington, DC, USA, 2011. IEEE Computer Society.

[10] R. J. Turetsky and D. P. Ellis. Ground-truth transcriptions of real music from force-aligned midi syntheses. *ISMIR 2003*, pages 135–141, 2003.

[11] S. Vantigodi and R. V. Babu. Human action recognition using motion capture data. In *National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics*. 2013.

[12] S. Vantigodi and R. V. Babu. Action recognition from motion capture data using metacognitive RBF network classifier. In *International Conference on Intelligent Sensors, Sensor Networks and Information Processing*. 2014.

[13] R. Vemulapalli, F. Arrate, and R. Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 588–595, June 2014.

[14] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *IEEE Conference on Computer Vision and Pattern Recognition,*, pages 1290–1297, June 2012.

[15] L. Xia, C.-C. Chen, and J. K. Aggarwal. View invariant human action recognition using histograms of 3d joints. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 20–27, 2012.

[16] X. Yang and Y. Tian. Eigenjoints-based action recognition using naive-bayes-nearest-neighbor. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 14–19, June 2012.