2014 IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO)
September 11-13, 2014. Evanston, Illinois, USA

3D Human Action Segmentation and Recognition using Pose Kinetic Energy

Junjie Shan and Srinivas Akella

***Abstract*— Human action recognition is challenging, due to large temporal and spatial variations in actions performed by humans. These variations include significant nonlinear temporal stretching. In this paper, we propose an intuitively simple method to extract action templates from 3D human joint data that is insensitive to nonlinear stretching. The extracted action templates are used as the training instances of the actions to train multiple classifiers including a multi-class SVM classifier. Given an unknown action, we first extract and classify all its constituent atomic actions and then assign the action label via an equal voting scheme. We have tested the method on two public datasets that contain 3D human skeleton data. The experimental results show the proposed method can obtain a comparable or better performance than published state-of-the-art methods. Additional experiments also demonstrate the method works robustly on randomly stretched actions.**

I. Introduction

Action recognition has many applications in robotics and computer vision [26],[17],[15] including healthcare monitoring and human-robot interaction. Recognizing human actions from RGB image sequences has been a long-standing challenge [32],[2], partly due to the fact that it is difficult to robustly track humans and extract action descriptors from RGB images. The recent advent of inexpensive depth cameras such as the Microsoft Kinect sensor has alleviated the problem as depth images make tracking and extraction less difficult. However, if body parts are not correctly labeled, precisely recognizing actions from depth images is still challenging. Fortunately, extracting human skeleton coordinates from a single depth image can be efficiently done in real time [30]. While conventional motion capture (MoCap) techniques employ expensive reflective markers or intrusive inertial sensors to provide 3D marker coordinates, a depth sensor based MoCap system has a much lower cost and will be much simpler to set up. Therefore, our focus is on developing an algorithm that can recognize human actions from 3D joint coordinates in a fast and robust manner.

Still, due to the varied nature of human actions, it is not trivial to extract expressive features from 3D coordinates to recognize actual human actions. Specifically, there are several difficulties in the temporal domain we need to address, such as random pauses, nonlinear stretching, and repetitions. Due to these challenges, if we directly use the raw 3D coordinates as features without preprocessing or feature selection, the intra-class variance will be very large, making recognition difficult. Although algorithms like HMM and MEMM have had some success in classifying sequence-based human actions, their optimization can be easily undermined by local maxima and multiple parameters. Additionally, they are not specifically designed to tackle challenges like temporal stretching. Therefore, a fast and simple method to recognize human actions and that is robust to temporal stretching is necessary.

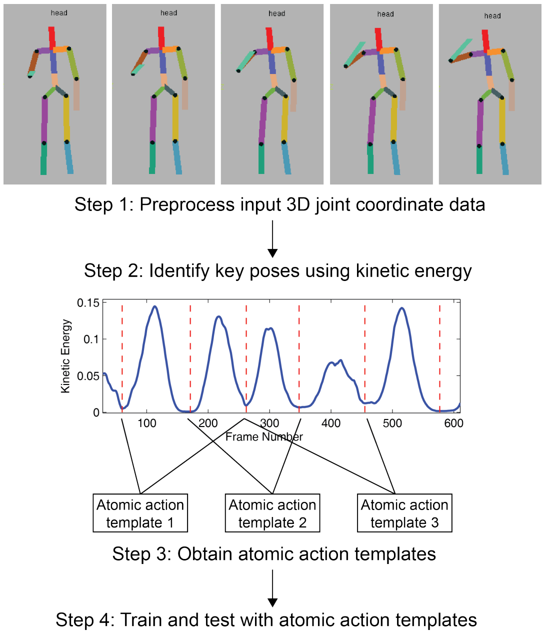
head
head
head
head
head
Step 1: Preprocess input 3D joint coordinate data
Step 2: Identify key poses using kinetic energy
Kinetic Energy
0.15
0.1
0.05
0
100
200
300
400
500
600
Frame Number
Atomic action template 1
Atomic action template 2
Atomic action template 3
Step 3: Obtain atomic action templates
Step 4: Train and test with atomic action templates

Fig. 1: Outline of the method to segment and extract poses for action recognition.

In this paper, we present a method to recognize human actions from 3D human skeleton coordinates. It consists of a sequence of steps to address the spatial and temporal challenges. The 3D skeleton data is normalized so that all poses have the same size, position, and orientation relative to the sensor. Then segmentation is performed on the pose sequence to extract key poses, which constitute the atomic action templates (Figure 1). This step eliminates the influence of nonlinear stretching and random pauses in the temporal domain, thus greatly reducing intra-class variance. The third step builds a classification model that can classify actions accurately and efficiently.

The remainder of this paper is organized as follows. Section II introduces related work, Section III describes the preprocessing of 3D skeleton data and the feature extraction. The classifier design is discussed in Section IV. Experimental results and analysis are presented in Section V. Section VI concludes with a summary and discussion of future work.

This work was partially supported by National Science Foundation Award IIS-1258335.

J. Shan and S. Akella are with the Department of Computer Science, University of North Carolina at Charlotte, Charlotte, North Carolina 28223, USA. jshan1@uncc.edu, sakella@uncc.edu

978-1-4799-6968-5/141$31.00 ©2014 IEEE

II. Related Work

A. Input Features

Various types of sensor data have been used for human action recognition. Early work on action recognition was done with RGB image and video data. Aggarwal and Ryoo's comprehensive survey [2] focuses on human action recognition in images and videos, and describes and compares the most widely used approaches. Action recognition and monitoring using inertial sensor data was explored by Maurer et al. [21]. However, these sensors are more intrusive and expensive than RGB cameras, and also cannot be set up and accessed as easily as them.

Going beyond RGB images and videos, the recent advent of inexpensive RGB-D sensors such as the Microsoft Kinect [1] that directly provide depth information has triggered an increasing interest in using depth images for pose estimation [30],[10] and action recognition [18],[31]. Depth images coupled with RGB images have more information available to discriminate different poses and actions, compared to RGB images. Chen et al. [7] summarize research on pose estimation and action recognition using depth images. Li et al. [18] used a sampling method to sample from 3D points in depth images to recognize actions. Wang et al. [33] extract Random Occupancy Patterns from depth sequences, use sparse coding to encode these features, and classify actions. Munaro et al. [22] developed a 3D grid-based descriptor from point cloud data and recognize different actions using nearest neighbors.

An alternative approach is to utilize 3D data on skeleton joint positions from the depth information. Given a single pose image, 3D joint positions of a human can be estimated [30]. Sung et al. [31] combine both the 3D skeleton information and depth images to detect and recognize human actions. Their feature set for a static pose has about about 700 elements, including joint positions and Histogram of Oriented Gradients (HOG) of RGB and depth images. Xia et al. [39] use 3D joint coordinates to describe the skeleton configuration using spherical angles computed to the joints from a reference point at the hip. They discretize the spherical space and use the histogram of 3D joint locations as features for a discrete HMM to perform activity recognition. Zhang and Tian [41] define a structure similarity in different actions and employ a set of SVM classifiers to recognize actions. Oreifej and Liu [25] extract body parts and joint coordinates from depth images, and then use the histogram of oriented 4D normals of body parts and joint coordinates to recognize actions.

In this paper, only 3D skeleton data estimated from depth sensors will be employed. We are interested in the recognition performance when only 3D coordinates of joints of humans are used. The advantage of using 3D coordinates is the smaller number of features, compared to depth images.

B. Classification Model

Human actions can be viewed as time series data. The approach to classification of time series data is slightly different from conventional pattern recognition problems, since training and testing samples may have different lengths. Usually, there are two classes of methods for time series recognition, sequence-based classification algorithms and classification using high-level feature extraction.

Among the sequence-based classification algorithms, dynamic time warping (DTW), hidden Markov models (HMM) and maximum-entropy Markov models (MEMM) have been used in many applications. DTW has been used to represent, recognize, and predict activities with the motion trajectory features [8],[11],[38],[29]. Calinon et al. [4], [5] employ HMMs to recognize and reproduce movement of humanoid robots. Piyathilaka and Kodagoda [27] use a Gaussian mixture model based HMM to recognize human activities using 3D positions of each skeleton joint as input features. Sung et al. [31] use a two-layer MEMM to recognize human actions from both 3D skeleton information and depth images.

However, if we extract high-level features from pose sequences, additional machine learning algorithms can be used [41][40][35]. Yang and Tian [40] used Accumulated Motion Energy (AME) computed from 3D skeleton joints and used a non-parametric Naive-Bayes-Nearest-Neighbor (NBNN) classifier. Wang et al. [34], [35] proposed an actionlet ensemble model using a temporal pyramid of Fourier transform coefficients of 3D joint positions and depth images, and tested it on five datasets.

This paper performs classification using high-level features extracted using our segmentation algorithm. Various segmentation techniques have been developed in computer vision [20], [37], [13], [16]; our approach is most closely related to Marr and Vaina's [20] state-motion-state shape representation for motion recognition. In this paper, we extract key poses using kinetic energy to segment sequences of human action data.

III. Our Approach

In this section, we first describe the preprocessing of 3D skeleton data. We then explain the extraction of temporal features of actions, a key aspect of the paper.

A. Preprocessing of 3D Skeleton Data

The (x, y, z) coordinates of human joints can be obtained from RGB-D sensors. It is therefore natural to use the coordinates of all J joints as a descriptor of a static *pose* P, i.e., a configuration of a skeleton. However, raw coordinates cannot be used directly, since they are dependent on variables such as the human subject's height, limb length, and orientation and position with respect to camera. Hence we preprocess the raw coordinates to make the descriptor of static pose insensitive to the aforementioned variables.

To normalize the 3D skeleton data, raw 3D coordinates are preprocessed by the following three steps:

1) **Translation**. To eliminate the effect of different camera positions, the origin of the coordinate system is translated to a predefined joint on the human skeleton, e.g., hip center.

2) **Rotation**. The static poses are rotated to a predefined orientation.
3) **Scaling**. After translation and rotation, we uniformly scale all poses into a fixed range using their height and shoulder width, reducing the influence of differences in human height and limb lengths.

The resulting coordinates are normalized. The coordinates of the joint selected as the origin are (0, 0, 0), and only $3(J-1)$ values are needed to represent a static pose.

For real datasets, additional preprocessing steps are needed due to sensor error and noise. A simple moving average is employed to smooth the data. In the presence of substantial noise, estimated coordinates provided by the depth camera may be corrupted. "Corrupted" poses are those poses that cannot be validated as feasible human poses; they usually lead to sharp discontinuities in motion. For example, when a subject is not in the view of depth camera, the estimated joint coordinates during this time may all be zeros, which may compromise the entire action sample. Therefore, it is necessary to detect corrupted poses. In our implementation, if a pose is slightly corrupted, we correct the pose based on geometric information. Otherwise, if the pose is heavily corrupted, it is simply discarded.

B. Identifying Key Poses using Pose Kinetic Energy

We next define key poses and pose kinetic energy. Afterward, we discuss the usage of pose kinetic energy for identifying key poses.

In a sufficiently long action sample with multiple repetitions of atomic actions, the movement of certain human joints show acceleration and deceleration alternately. This leads to changes in the total kinetic energy of all joints. Therefore, some special poses that have the minimal kinetic energy in a local neighborhood are good descriptors of an action type, because these poses identify extremal positions of action in high dimensional space. We refer to these poses as *key poses*. For example, one key pose in a drinking water action is where the human's glass-grasping hand changes its movement direction, from moving upward to moving downward. Key poses can be used to segment and recognize actions, as we demonstrate in this paper.

A human *action sample* S is an ordered sequence of poses $S=(P_1,P_2,\ldots,P_T)$, where the i^{th} pose $P_i \in \mathbb{R}^d$, T is the number of poses in the sequence, and d is the dimension of each pose. S can be viewed as a path from P_1 to P_T in $\mathbb{R}^d$ space.

The kinetic energy $E(P_i)$ at a pose P_i is defined to be the sum of the kinetic energy over all its d dimensions. That is,

$$E(P_i)=\sum_{j=1}^{d} E(P_i^j) \tag{1}$$

where P_i^j is the jth dimension of the pose P_i.

The kinetic energy is proportional to the square of velocity. We ignore the mass term in kinetic energy as it is not relevant. The velocity can be approximated in our case by considering finite differences of position divided by the sampling time interval ΔT, i.e.

$$v_i^j=\frac{P_i^j-P_{i-1}^j}{\Delta T} \tag{2}$$

$E(P_i^j)$, the kinetic energy of the jth dimension of pose P_i, can be computed as follows:

$$E(P_i^j)=\frac{1}{2}\left(\frac{P_i^j-P_{i-1}^j}{\Delta T}\right)^2 \tag{3}$$

Substituting Equation 3 into Equation 1, we obtain,

$$E(P_i)=\frac{1}{2}\sum_{j=1}^{d}(\frac{P_i^j-P_{i-1}^j}{\Delta T})^2 \tag{4}$$

The key poses are those stationary poses that have zero kinetic energy, which means a key pose P^* must satisfy:

$$E(P^*)=0 \tag{5}$$

However, since the sampling frequency may not be sufficiently high to capture the exact pose when the key pose has zero energy, the computed kinetic energy of a key pose may be slightly larger than zero. To make the algorithm more robust for real data, we change Equation 5 to:

$$E(P^*)<E_{minimal} \tag{6}$$

where $E_{minimal}$ is a parameter that we need to tune through experiments.

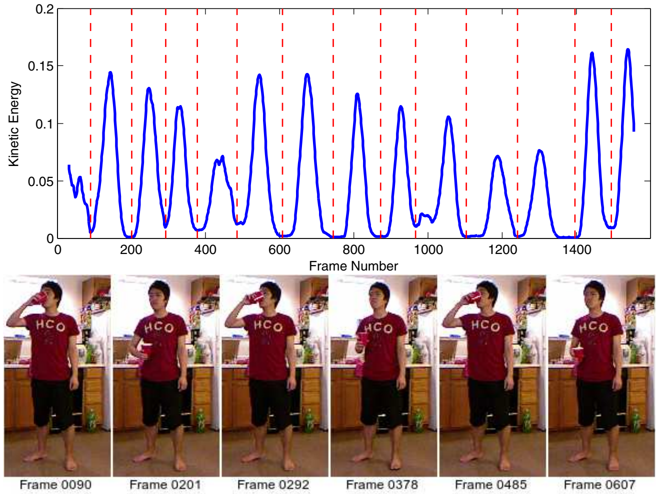

Fig. 2: Key poses identified from an example "drinking water" action sample from the Cornell dataset are indicated by the dashed vertical lines (top). A subset of the identified key poses are shown in the images (bottom).

Fig. 2 shows an example of extracted key poses using a "drinking water" action sample from the Cornell Activity dataset [31]. The "drinking water" atomic action has been-repeated seven times, each repetion of which differs from the others in temporal duration and spatial path. Despite the obvious challenges posed by variations in the magnitude and speed of each atomic action, the presented method based on pose kinetic energy successfully identifies all the key poses.

The key poses extracted by the method should be insensitive to the temporal stretching commonly seen in real actions. In Section V-A.4, we will demonstrate the robustness of the method with experiments that show the key poses can overcome distortion due to random temporal stretching.

C. Atomic Action Templates

After key poses are identified, we can use them to compose the *atomic action templates*, a spatiotemporal representation. One common feature representation in information retrieval and computer vision is a bag-of-words model [19]. However, since the relative order of poses is critical to describe human actions, we instead define an atomic action template as an n-tuple consisting of n poses in temporal order.

Atomic action templates will more closely approximate samples as n increases. However, n should not be too large, otherwise it will make the feature representation less generalized and increase computation. In this paper, we chose n to be 5 after extensive experiments. In one atomic action template, the first, third and fifth pose are three key poses directly obtained from pose kinetic energy, while the second pose is temporally in the middle of the first and the third poses, and the fourth pose is midway between the third and fifth poses. So one atomic action template is a feature vector of $3n(J-1)$ scalar values. There is a template originating at each key pose, with overlap between successive templates.

The underlying assumption of this method is that an action is defined by the presence of a few key poses, independent of the majority of the intermediate poses and the speed of the motion between poses.

In the datasets we have tested, the action samples have different numbers of poses as well as different numbers of repetitions. One or more atomic action templates can be extracted from an action sample. In the training phase, all the atomic action templates belonging to the same action type are training instances of the action type. In the testing phase, a testing action sample is processed using the presented pipeline (Fig. 1). First, the identified one or more atomic action templates will be extracted from the testing action sample; then, the label of each atomic action template will be determined by the classifier we have trained; finally, the class label of the testing action sample will be assigned via an equal voting scheme.

IV. Classification Models

After the features for an action sample are extracted, they are fed into a classification model. To test our feature representation more thoroughly, we employ four classification models. HMM is chosen to be the baseline classification model, as a typical generative model and sequence-based classification algorithm. In addition, we used K-Nearest Neighbor (KNN), Support Vector Machine (SVM), and Random Forest (RF). We now discuss their settings.

- HMMs [28],[3] have found wide applicability in many areas such as speech recognition. A HMM first has to be trained for each action type. Given a set of observation sequences, we can learn the parameters of HMMs using the Expectation Maximization (EM) algorithm [28],[3]. Given an action sample A_j, which consists of a series of L observed poses, $o_1, o_2, \ldots, o_L$, we can calculate the probability of the action A_j for the ith action type T_i, that is $P(A_j|T_i)$. In general, an HMM is defined by three elements: the prior distribution for initial states π, the transition matrix $M(S_{t+1}|S_t)$, and the emission matrix $P(o_t|S_t)$ where S_t is the hidden state at time t. We use the Gaussian distribution to model the coordinate values. Hence, the goal of training the HMM is to learn all these parameters.
- K-Nearest Neighbor (KNN) is a lazy instance-based classifier without training. When using KNN, all we need to do is to provide a distance function to measure proximity. Since the time complexity of predicting a sample point with KNN is $O(N)$, where N is the number of training samples, it is not recommended for large N. Since the number of training samples and testing samples in the datasets we have tested is very small, roughly 100 in each case, using KNN is a viable option. We let K be 1 and choose the canonical Euclidean Distance function to measure proximity.
- Support Vector Machines (SVM) try to find a hyperplane that can separate different classes. When combined with kernel tricks, SVM shows good performance on nonlinear classification problems. Unlike KNN, SVM does not need to store all samples after training, and it is faster in classifying a new sample. However, selecting optimal kernel functions and tuning parameters are the main challenges of applying SVM. Since the number of features for a sample is very high (1000+), we found that the linear kernel is a better choice than the radial basis function (RBF) kernel.
- Random Forests (RF) is a type of ensemble classifier that consists of multiple decision trees. When using RF, few parameters need to be chosen. The number of decision trees is 200 in our experiments using RF.

V. Experimental Results

In this section, we describe the experimental settings and datasets we used, and our test results. Finally, we compare our results with state-of-the-art results.

To evaluate the effectiveness of our method for recognition of 3D human actions, we used two public datasets: Cornell Activity dataset [31] and MSR Action3D dataset [18]. To compare our results with previous work, we use the same experimental settings.

The parameters in our method are chosen as follows. The size n of the atomic action template is determined by the complexity of atomic actions, which depend on the number of distinct key poses present. More complex atomic actions may need larger n, with a resulting increase in computation. To find the optimal value of n, we tested values from 1 to 20 on the Cornell activity data and MSR Action3D data. We found n of 5is sufficiently large to obtain good recognition results while keeping computation efficient. $E_{minimal}$ is determined by the noise level of the 3D coordinates. In our experiments, we let $E_{minimal} = 10^{-2}\ m^2/s^2$. All experiments were run using Matlab, using software packages including LIBSVM [6] and randomForest-matlab [14].

A. Cornell Activity Dataset

1) Data Set: The Cornell Activity Dataset [31] focuses on realistic actions from daily life. The dataset was collected using the Microsoft Kinect sensor and actions were performed by four different human subjects, two males and two females. Three of the subjects use the right hand to perform actions, one of them uses the left hand. There are 12 type of actions in the dataset, which are: "talking on the phone", "writing on whiteboard", "drinking water", "rinsing mouth with water", "brushing teeth", "wearing contact lenses", "talking on couch", "relaxing on couch", "cooking (chopping)", "cooking (stirring)", "opening pill container", and "working on computer".

2) Experimental Setup: Since the actions to be recognized may come from subjects whose actions are in the training set, or from subjects whose actions are not in the training set, each experiment must be conducted on both cases. We adopt the naming convention from [31], calling them the Have Seen setting and New Person setting respectively. We follow the exactly same experiment setting in [31] and run the "Have Seen" and "New Person" setting separately.

In the raw Cornell Dataset, most action samples actually contain several repetitions of an atomic action. Segmentation (Section III) is performed on the action samples. Beyond the repetitions, there is another issue, the left-handedness and right-handness of subjects. Three of the human subjects use the right hand to perform actions, one of them uses the left hand. Semantically, "drinking water with right hand and "drinking water with left hand are equivalent because both belong to "drinking water" action type. However, in the feature space, they may not be close. To address the issue, we double every action type to a right-hand version and a left-hand version by creating mirrored copies. We train and test on the left-handed and right-handed action types separately and merge the results when computing the average precision and recall.

All the results are obtained under the same settings for all four classifiers. The performance numbers we report are obtained consistently.

3) Comparison with Prior Work: The precision/recall values of "Have Seen" and "New Person" setting are reported in Table I. In "Have Seen" setting, the precision/recall reach nearly 100%. The overall precision/recall of "Have Seen" is higher than that of "New Person", which is consistent with the fact that inter-person variations make it significantly more difficult to recognize actions from a new person. Our discussion will focus on the "New Person" setting because in many real application scenarios, most human actions to be recognized will be from "new" subjects.

Table I also shows that the RF, SVM, and KNN classifiers obtain nearly the same recognition performance. This demonstrates also that the extracted feature representation is sufficiently discriminative.

Note that we use only extracted features from the 3D coordinates, while most other methods also used features extracted from depth images. Table II compares the performance of our method against a number of state-of-the-art methods. The presented method outperforms previous methods by achieving 93.8%/94.5% precision/recall in the "New Person" setting.

TABLE I: Precision/Recall (%) of multiple classifiers on Cornell dataset.

Setting	HMM		RF		SVM		KNN	
	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
Have Seen	96.8	98.9	99.8	99.8	**100.0**	**100.0**	98.8	99.0
New Person	74.2	82.5	87.9	91.2	**93.8**	**94.5**	93.3	95.2

TABLE II: A comparative summary of the performance of prior methods and our method on the Cornell Activity dataset. (NA: Not available.)

Algorithm	New Person		
	Prec. (%)	Rec. (%)	Accy. (%)
MEMM [31]	67.9	55.5	NA
SSVM [16]	80.8	71.4	NA
Kinematic Feature [41]	86	84	NA
Sparse Coding [23]	NA	NA	65.32
Eigenjoints [40]	71.9	66.6	NA
HMM+GMM [27]	70	78	NA
Image Fusion [24]	75.9	69.5	NA
Depth Images Segment [12]	78.1	75.4	NA
Actionlet [35]	NA	NA	74.70
Spatiotemporal Interest Pt. [42]	93.2	84.6	NA
Probabilistic Body Motion [9]	91.1	91.9	NA
Our Algorithm	**93.8**	**94.5**	**91.9**

4) Robustness under Random Stretching: Our algorithm performs well on actions with temporal variations. The quality of atomic action templates depends on the extracted key poses. To test the algorithm's robustness when actions have greater temporal variation, we designed an experiment to evaluate the ability of the algorithm to identify key poses even with significant temporal stretching. We use the same experiment setting as in the previous section. The only difference is that we apply random temporal stretching to the original actions.

The original actions are randomly stretched as follows. Given a raw action sample in the Cornell Activity dataset, we first randomly split the action into a random number of short segments. Second, we randomly compress or lengthen each segment temporally. Lastly, we concatenate all the modified segments in their original order. All the training and testing are conducted on the actions that have been randomly stretched. Fig. 3 shows an example of random stretching. It can be easily seen that all the atomic action templates have been strongly stretched in a nonuniform manner.

The result of extracting the key poses is shown in Fig. 4. Comparing the result with Fig. 2, we can confirm that the extracted key poses are nearly the same, which demonstrates that the method to identify key poses is very robust to temporal stretching.

We next describe the results of testing recognition performance of actions under random stretching. Table III shows the resulting precision and recall using the atomic action template when SVM is selected as the classifier. The precision/recall values in the "Have Seen" setting are 98.4%/98.8%, and the precision/recall values in the "New Person" setting are 94.2%/95.4%. In comparison to the performance on original actions without random stretching, the performance on randomly stretched actions is degraded by a small amount on "Have Seen" examples and in fact improves on "New Person" examples. Again, the result demonstrates that the presented method to identify key poses and use atomic action templates is robust to temporal stretching.

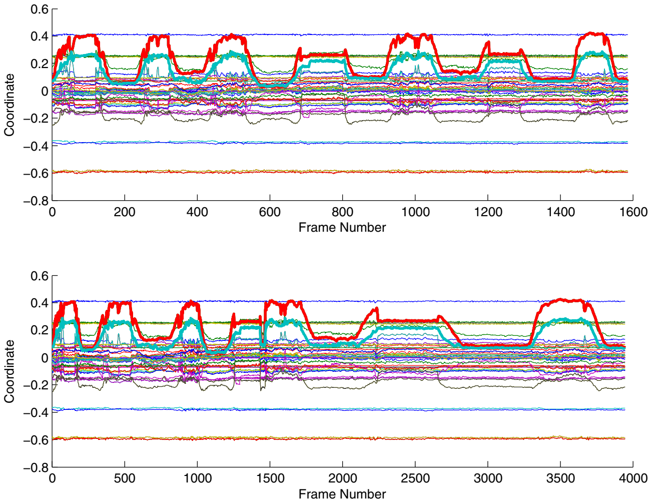

Fig. 3: Plots of all joint coordinates of a "drinking water" action sample (top) and its randomly stretched version (bottom).

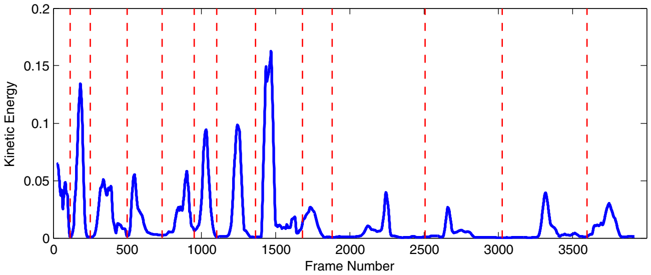

Fig. 4: Plot of kinetic energy with key poses identified on the randomly stretched "drinking water" action sample of Figure 3.

TABLE III: Prec./Rec. (%) of SVM on randomly stretched Cornell dataset.

Setting	SVM	
	Prec.	Rec.
Have Seen	98.4	98.8
New Person	94.2	95.4

B. MSR Action3D Dataset

1) Data Set: The MSR Action3D dataset [18] contains 20 action types performed by ten human subjects. The data is collected with a depth camera similar to the Microsoft Kinect sensor. The action types are categorized into three subsets. There are 567 action samples in the dataset, of which 557 were usable after removing 10 heavily corrupted action samples.

2) Experimental setup: To evaluate the action recognition performance, we conducted experiments with three different sets of training and testing samples, using the same settings as in [18]. We concentrate our discussion on the "New Person" ("Cross Subject Test" in [18]) setting.

3) Comparison with other methods: The evaluation of our method is shown in Table IV. In Cross Subject Test, the best average accuracy obtained is 84.0% when Random Forest is employed as the classifier. The result is not directly comparable to other state-of-art algorithms using both depth image and coordinates. However it is better than the state-of-the-art technique without depth images by 1.7%. There are several possible reasons why our algorithm does not get the same performance on the MSR Action3D dataset as on the Cornell Dataset. Firstly, inter-person variance of MSR Action3D is larger than that of Cornell Dataset. The Cornell dataset is recorded with only four human subjects and experiments are conducted by training on three subjects and testing on the remaining one subject. However, there are ten subjects in MSR Action3D dataset, with training on five subjects and testing on the other five subjects. Secondly, the 3D skeleton data extracted from MSR Action3D data is more noisy, with more corrupted and occluded poses. The proposed segmentation algorithm cannot completely recover corrupted and occluded poses without using additional information, such as the depth image. Thirdly, most action samples in MSR Action3D dataset do not have identifiable key poses, i.e., poses with zero kinetic energy.

TABLE IV: Average accuracy (%) of multiple classifiers on the MSR Action3D Dataset in Cross Subject setting.

	HMM	RF	SVM	KNN
Average Accuracy (%)	55.8	**84.0**	83.6	75.2

TABLE V: A comparative summary of the performance of prior methods and our method on the MSR Action3D dataset.

Method	Used Depth Images	Average Accuracy (%)
Bag of 3D Points [18]	Yes	74.7
Histogram of 3D Joints [39]	Yes	78.9
Random Occupancy Pattern [33]	Yes	86.2
Actionlet Ensemble [35]	Yes	88.2
HON4D + Ddesc [25]	Yes	88.9
MMTW [36]	Yes	92.7
Eigenjoints [40]	No	82.3
Our Algorithm	**No**	**84.0**

VI. Conclusion

In this paper, we presented a method to extract features from 3D joint coordinates in human action data and then recognize actions using the extracted features. To extract good features, multiple steps are taken to address the large intra-class variances, including pose normalization in the spatial domain, and segmentation in the temporal domain. Atomic actions consisting of key poses are demonstrated to show good discriminative power with multiple classifiers. The experimental results show the proposed method can outperform published methods while using less input data. In our future work, we plan to develop new methods to address the challenge of improving recognition performance when poses are corrupted and heavily occluded. Further, we will evaluate our method on real-time streaming action data.

References

[1] Microsoft Kinect for Windows. `http://www.microsoft.com/en-us/kinectforwindows/`. Accessed: 2013-05-21.

[2] J. K. Aggarwal and M. S. Ryoo. Human activity analysis: A review. *ACM Computing Survey*, 43(3):16, 2011.

[3] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[4] S. Calinon and A. Billard. Recognition and reproduction of gestures using a probabilistic framework combining PCA, ICA and HMM. In *International Conference on Machine Learning (ICML)*, pages 105–112, 2005.

[5] S. Calinon, F. Guenter, and A. Billard. On learning, representing, and generalizing a task in a humanoid robot. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 37(2):286–298, 2007.

[6] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[7] L. Chen, H. Wei, and J. M. Ferryman. A survey of human motion analysis using depth imagery. *Pattern Recognition Letters*, 34:1995–2006, 2013.

[8] T. Darrell and A. Pentland. Space-time gestures. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 335–340, 1993.

[9] D. R. Faria, C. Premebida, and U. Nunes. A probabilistic approach for human everyday activities recognition using body motion from RGB-D images. In *IEEE RO-MAN'14: IEEE International Symposium on Robot and Human Interactive Communication*, 2014.

[10] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun. Real-time human pose tracking from range data. In *European Conference on Computer Vision*, pages 738–751, 2012.

[11] D. M. Gavrila and L. S. Davis. Towards 3-d model-based tracking and recognition of human movement: A multi-view approach. In *International workshop on automatic face-and gesture-recognition*, pages 272–277, 1995.

[12] R. K. Gupta, A. Y. S. Chia, and D. Rajan. Human activities recognition using depth images. In *ACM Multimedia*, pages 283–292, 2013.

[13] M. Hoai, Z.-Z. Lan, and F. De la Torre. Joint segmentation and classification of human actions in video. In *2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3265–3272. IEEE, 2011.

[14] A. Jaiantilal. Classification and regression by randomforest-matlab. Available at `https://code.google.com/p/randomforest-matlab`, 2009.

[15] O. C. Jenkins and M. J. Matarić. Deriving action and behavior primitives from human motion data. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2002)*, volume 3, pages 2551–2556, Lausanne, Switzerland, Oct 2002.

[16] H. Koppula, R. Gupta, and A. Saxena. Learning human activities and object affordances from RGB-D videos. *International Journal of Robotics Research*, 32(8):951–970, 2013.

[17] V. Krüger, D. Herzog, S. Baby, A. Ude, and D. Kragic. Learning actions from observations. *IEEE Robotics and Automation Magazine*, 17(2):30–43, 2010.

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

[19] C. D. Manning, P. Raghavan, and H. Schutze. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK, 2008.

[20] D. Marr and L. Vaina. Representation and recognition of the movements of shapes. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 214(1197):501–524, 1982.

[21] U. Maurer, A. Smailagic, D. P. Siewiorek, and M. Deisher. Activity recognition and monitoring using multiple sensors on different body positions. In *International Workshop on Wearable and Implantable Body Sensor Networks*, pages 113–116, 2006.

[22] M. Munaro, G. Ballin, S. Michieletto, and E. Menegatti. 3D flow estimation for human action recognition from colored point clouds. *Biologically Inspired Cognitive Architectures*, 5:42–51, 2013.

[23] B. Ni, P. Moulin, and S. Yan. Order-preserving sparse coding for sequence classification. In *European Conference on Computer Vision (2)*, pages 173–187, 2012.

[24] B. Ni, Y. Pei, P. Moulin, and S. Yan. Multilevel depth and image fusion for human activity detection. *IEEE Transactions on Cybernetics*, 43(5):1383–1394, Oct. 2013.

[25] O. Oreifej and Z. Liu. HON4D: Histogram of oriented 4D normals for activity recognition from depth sequences. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 716–723, 2013.

[26] C. Ott, D. Lee, and Y. Nakamura. Motion capture based human motion recognition and imitation by direct marker control. In *2008 8th International Conference on Humanoids*, pages 399–405, Daejeon, S. Korea, Dec. 2008.

[27] L. Piyathilaka and S. Kodagoda. Gaussian mixture based HMM for human daily activity recognition using 3D skeleton features. In *2013 8th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pages 567–572, 2013.

[28] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, Feb. 1989.

[29] Z. Shao and Y. F. Li. A new descriptor for multiple 3D motion trajectories recognition. In *IEEE International Conference on Robotics and Automation*, pages 4734–4739, Karlsruhe, Germany, May 2013.

[30] J. Shotton, T. Sharp, A. Kipman, A. W. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124, 2013.

[31] J. Sung, C. Ponce, B. Selman, and A. Saxena. Unstructured human activity detection from RGBD images. In *IEEE International Conference on Robotics and Automation*, pages 842–849, 2012.

[32] P. K. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1473–1488, 2008.

[33] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu. Robust 3d action recognition with random occupancy patterns. In *European Conference on Computer Vision (2)*, pages 872–885, 2012.

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

[35] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Learning actionlet ensemble for 3D human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(5):914–927, May 2014.

[36] J. Wang and Y. Wu. Learning maximum margin temporal warping for action recognition. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2688–2695, Dec 2013.

[37] D. Weinland, R. Ronfard, and E. Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding*, 115(2):224–241, 2011.

[38] S. Wu and Y. F. Li. On signature invariants for effective motion trajectory recognition. *International Journal of Robotics Research*, 27(8):895–917, 2008.

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

[40] X. Yang and Y. Tian. Effective 3D action recognition using EigenJoints. *Journal of Visual Communication and Image Representation*, 25(1):2–11, Jan. 2014.

[41] C. Zhang and Y. Tian. RGB-D camera-based daily living activity recognition. *Journal of Computer Vision and Image Processing*, 2(4), Dec. 2012.

[42] Y. Zhu, W. Chen, and G. Guo. Evaluating spatiotemporal interest point features for depth-based action recognition. *Image and Vision Computing*, 32(8):453–464, 2014.