SPARSE REPRESENTATION OF SKELETAL TREE FOR ACTION RECOGNITION

Tzu-Yang Chen (陳子揚), Cheng-Yin Liu (柳成蔭) Chiou-Shann Fuh (傅楸善)
Li-Chen Fu (傅立成)

Dept. of Computer Science and Information Engineering,
National Taiwan University, Taiwan

E-mail: hsnu9722@hotmail.com

ABSTRACT

Recognizing human action in a video has a wide range of applications, such as surveillance system and human-robot interaction. However, it is a challenging task due to intra class variation and inter-class similarity. It usually requires high-dimensional feature vector to precisely describe the motion and posture of human that it may be hard to run in real-time. In this paper, we propose a novel and easy-to-implement representation for human action which can generally describe the motion of human skeleton over time. It can run in real-time for real world application and achieve reasonable performance.

1. INTRODUCTION

Kinect sensor is so popular nowadays due to its efficiency and effectiveness to acquire depth and human action information in real environment. In fact, there is a long history of sensors which capture human action. At first, we use wearable sensors such as Motion capture. However this kind of sensor is intrusive that we need to wear a number of sensors on our own body. Later we use hand-held sensor such as Wii. Although we only need hold one sensor, it is still clumsy and the motion information is incomplete. Until the emergence of Kinect, with this sensor, we can robustly extract the skeleton of human in a much more natural way that recognizing an action of human becomes more feasible and easier than before [1]. Fig. 1 shows the evolution of sensors to capture human motion. Fig. 3 illustrates the twenty joints captured by Kinect. Moreover, there are many advantages of extracting features on human skeleton captured by Kinect when compared with traditional RGB features. First, depth camera can much protect the privacy of user that it does not capture the appearance of people. Next, features of human skeleton can be more tolerant of variation of viewpoint that we can construct the coordinate system directly on human body. Third, since depth sensor does not capture appearance, its feature is appearance-invariant. Due to those advantages, our objective is therefore to design a skeleton descriptor to complete a system of action recognition that can tolerate challenging variation such as scale, viewpoint, and so on. Furthermore, many current approaches of action recognition are not able to run in real-time because of high-dimensional feature. As a result, we also require our system to be not only easy to implement but also efficient.

2. RELATED WORK

Nowadays, due to its wide applications there has

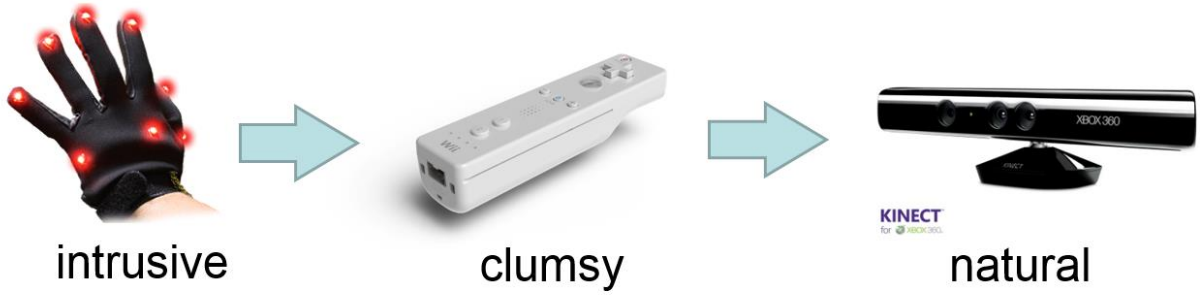

Fig. 1: The evolution of sensors which capture human motion.

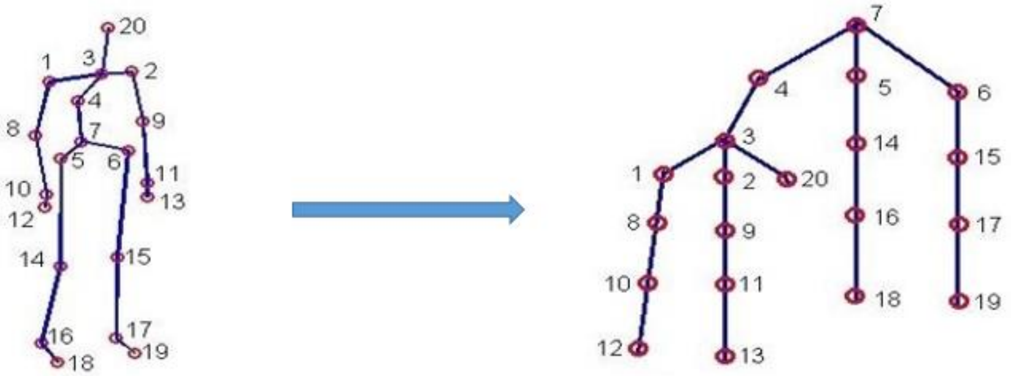

Fig. 2: Transform human skeleton into skeletal tree.

been a great deal of research working on action recognition system. Some of them directly apply or modify the well-known dense local features which are widely used in computer vision area for single image detection or classification, the features includes scale-invariant feature transform (SIFT) [2], speeded up robust features (SURF) [3], histogram of oriented gradients (HOG) [4], histogram of optical flow (HOF) [5], local binary pattern (LBP) [6] and motion boundary histogram (MBH) [7]. Ciptadi et al. [8] clusters optical flow into different words and use histogram to represent the motion of human action. Wang et al. [9] apply dense trajectories and eliminate background motion to represent video. Some work even extend those feature into 3-dimension to fit the property of video like HOG3D [10], HOF3D [11] and 3D SIFT [12]. Some work, on the contrary, designs the entire feature. HON4D [13] uses histogram to describe the distribution of the surface normal orientation in the 4D space of time, depth, and spatial coordinates. Wu et al. [14] describes the spatio-temporal context distribution of interest points. Lu et al. [15] use τ tests to construct a binary descriptor which is robust to occlusion and data corruption. Moreover, some work not only extract features from RGB or depth signal but also combine them with higher level representation such as a bag-of-features (BOF) framework [16] and sparse coding [17] to form new feature vector. Also, there is another type of features which focus on describing the motion or posture of human skeleton and our work belong to this type. Besides, the skeleton data can be either derived from RGB camera, Kinect or motion capture. Zia et al. [18] use the joint angle to construct a 3D model of human body which the skeleton is computed from RGB image. On the other hand, Ohn-Bar and Trivedi [19] and Sequence of the Most Informative Joints (SMIJ) [20] use the skeleton which is derived from Kinect. Vemulapalli et al. [21] explicitly models the 3D geometric relationships between various body parts and describe human motion as curves in this Lie group. In addition, some work also use deep learned feature [22] [23] . In these deep learning framework, they build a hierarchical structure and directly learn features from data. Although high computing cost, they usually can achieve great performance. in addition to the wide choice of features, there are also different classifier to use which includes Support Vector Machine (SVM), Hidden Markov Model (HMM) [24], K-nearest neighbor [25], decision tree and so on.

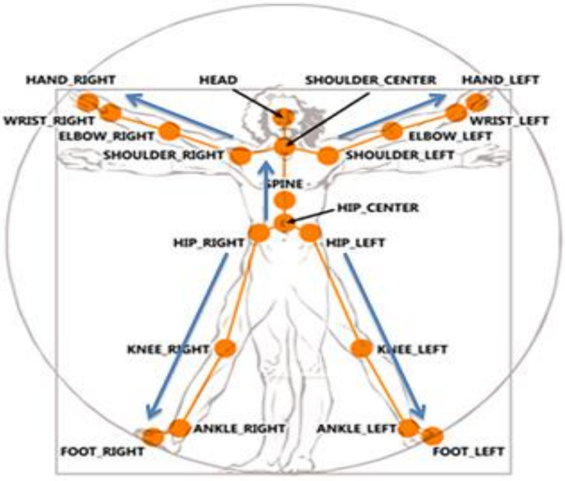

Fig. 3: The twenty joints captured by Kinect.

3. METHOD

We assume that human action mainly comprises series of joint motion over time. Based on this assumption, the basic idea is therefore to design a feature which can precisely describe the motion of each joint in human skeleton. And the feature is able to describe both temporal and spatial information of the joints. Then we will measure the similarities between training and testing data for classification. The method is composed of two main parts that we will illustrate below.

3.1 Relative Angle of Skeletal Tree

At first, as the skeleton is extracted by Kinect, we transfer the skeleton data into a tree where the root is the hip center joint. For each successive joint pair in the skeletal tree, it will build a parent-child pair in the skeletal tree and the joint which is closer to the hip center joint will become the parent node. Fig. 2 shows the corresponding relationship between human skeleton and skeletal tree. Second, inspired by [26] and [12], we use a spherical coordinate system to compute the azimuth and elevation angle of each joint; however, we design our own definition of the spherical coordinate system. For each joint, we set the center of the coordinate system as its parent node in the skeletal tree. The x-axis is the vector between right hip and left hip joint. The z-axis is defined as the vector which is normal to the line constructed by right and left hip joint and pass through the spine joint. Then, we acquire the y-axis by computing the cross product of x-axis and z-axis. Fig. 4 visualize the definition of axis direction in our spherical coordinate system. Once the axis of spherical coordinate system is defined, we can construct the spherical coordinate system and use it to compute the relative angle of each joint with respect to its parent node. Fig. 5 shows the coordinate system where the blue plane is constructed by X-axis and Y-axis and the light yellow plane is constructed by X-axis and Z-axis. Then, we project the child node onto both blue and light yellow plane that we can get the azimuth and elevation angle. Notice that we only extract the angle while abandon the radius information, the distance from child node to parent node, thus our representation is scale invariant. Third, we accumulate the angle of joint over time to form a vector that we can visualize the vector as a curve. As there are azimuth and elevation angle, each joint can construct two curves.

Moreover, we observed that some joints in the human skeleton are redundant for action recognition. For example, hand joint and wrist joint are so close to each other that we just need one of them to describe the

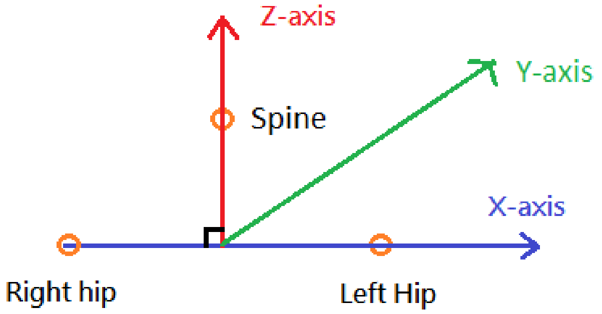

Fig. 4: The definition of axis in our spherical coordinate system. The Y-axis is derived from cross product of X-axis and Y-axis.

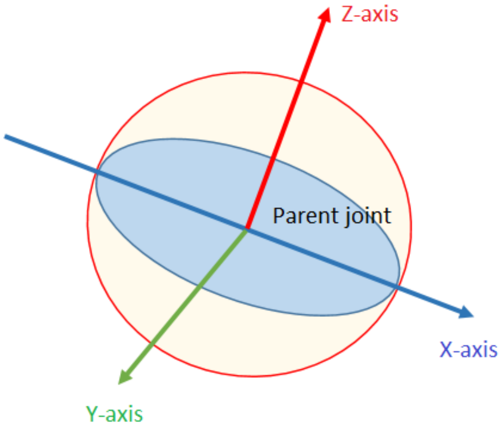

Fig. 5: The visualization of our spherical coordinate system.

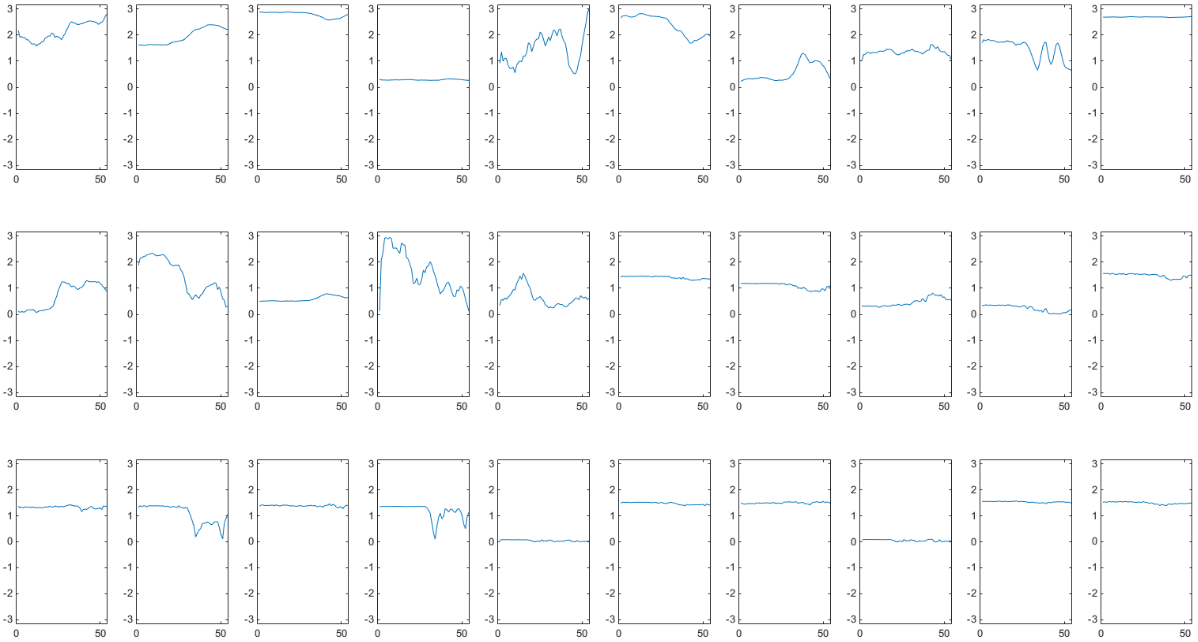

Fig. 6: One example of the thirty joint angles derived from a single video clip. X-axis denotes time domain and Y-axis denotes angle value.

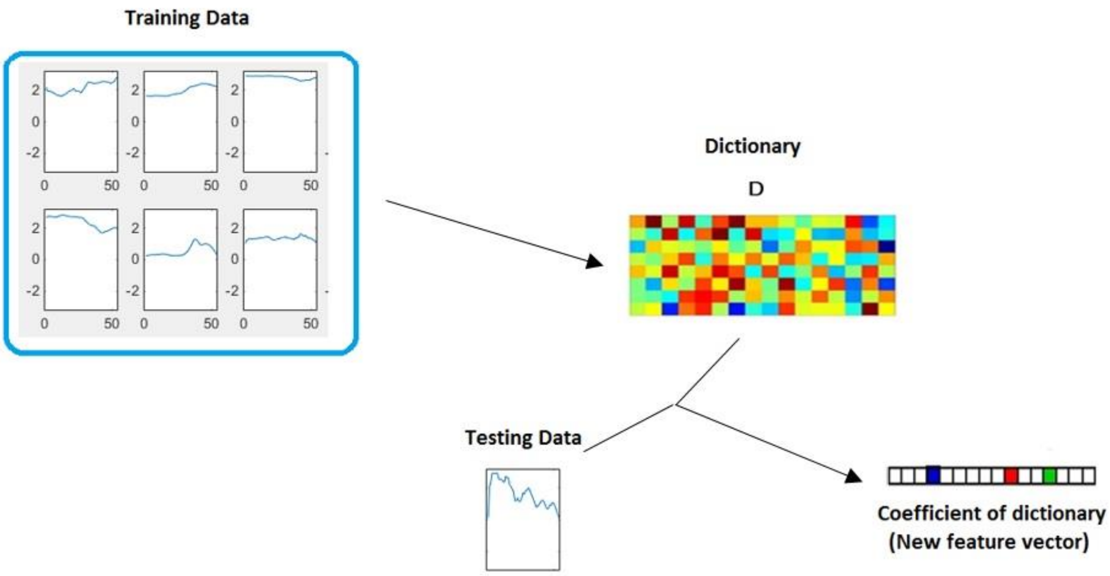

Fig. 7: Sparse representation in our system.

motion of the hand. Thus we decided to use only fifteen joints in the skeleton captured by Kinect. They include head, shoulder center, left and right shoulder, left and right elbow, left and right wrist, spine, hip center, left and right hip, left and right knee, left and right ankle. There are totally fifteen joints that will be used in the feature extraction. Fig. 6 shows some examples of azimuth angle curve of one video. Finally, there will be thirty angle curves for each action video clip.

3.2 Sparse representation of Joint angle

Although we can directly use the relative joint angle representation as our feature to classify action, we still want to extend this basic approach to higher level feature that can achieve better performance. We then use sparse coding method [10] which is usually applied in many deep learning applications as our second level representation of joint angle. We briefly go through the process of sparse coding here. For a finite training set of signals X = [x_1 ,…., x_n] in $R^{m \times n}$, our goal is to optimize the empirical cost function below

$$f_n(D) = \frac{1}{n}\sum_{i=1}^{n} l(x_i, D) \qquad (1)$$

where D in $R^{m \times k}$ is the dictionary and l is the loss function such that $l(x, D)$ will be small if D is good at

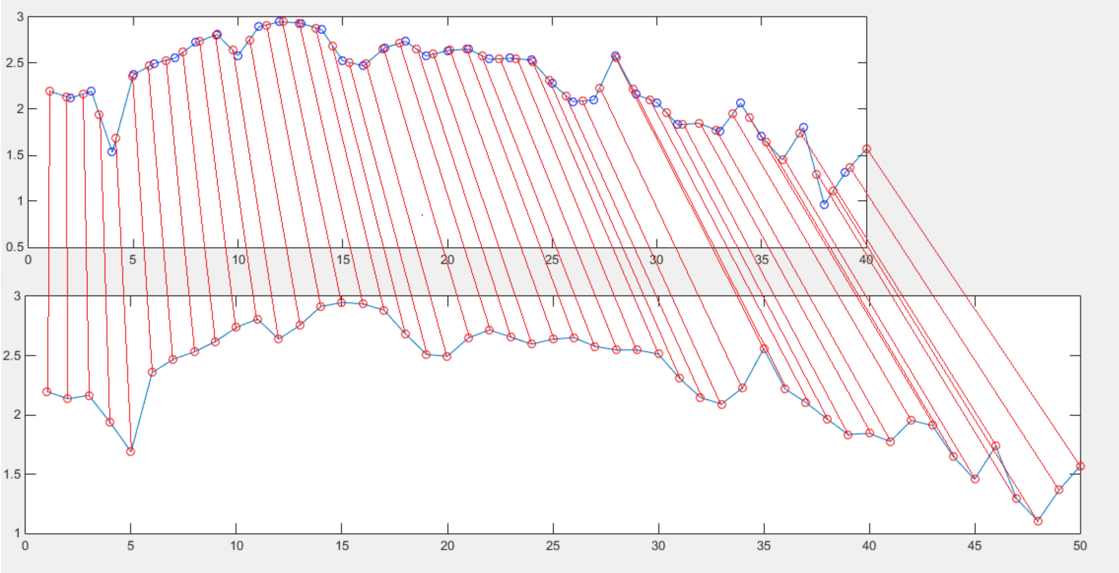

Fig. 8: An example that normalizes a video clip with 40 frames to 50 frames. The up-side is the original data which blue circles denote the 40 original sample points and the red points denote 50 new sample points. The down-side is the video clip after normalization.

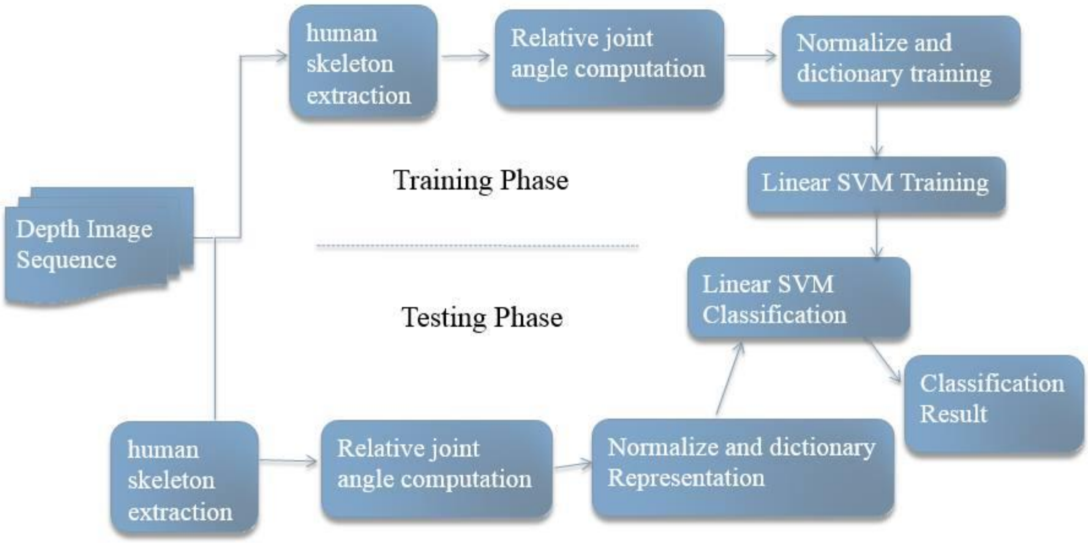
human skeleton extraction
Relative joint angle computation
Normalize and dictionary training
Linear SVM Training
Training Phase
Depth Image Sequence
Testing Phase
Linear SVM Classification
Classification Result
human skeleton extraction
Relative joint angle computation
Normalize and dictionary Representation

Fig. 9: System flow diagram

representing the input data x. We then define $l(x,D)$:

$$l(x,D)=\min_{\alpha\in R^k}\frac{1}{2}\|x-D\alpha\|_2^2+\lambda\|\alpha\|_1 \qquad (2)$$

where λ is a regularization parameter. Therefore, our empirical cost function becomes:

$$\min_{D\in C,\alpha\in R^{k\times n}}\frac{1}{n}\sum_{i=1}^{n}(\frac{1}{2}\|x-D\alpha\|_2^2+\lambda\|\alpha\|_1) \qquad (3)$$

C is the set of possible dictionary under certain constraint and we try to optimize this formula to get our dictionary. Since the dictionary is established, we can use (2) to transform each input signal into a vector of coefficient α with respect to dictionary D that it become a new feature vector.

In our experiment, we first construct a dictionary which is trained from all the angle curves in the training data, that is, each video in the dataset can provide thirty training data for the dictionary. Next, for each joint angle curves, we transform it into a sparse linear combination of words in the dictionary. There are two main advantages of doing so. First, sparse coding approach can use only a few of words which is more representative to represent original feature so it can be more general and avoid overfitting. Moreover, sparse coding use sparse vector structure to represent data and there are only eight words in the dictionary during the experiment, that is, we need just eight dimension sparse vector to describe each joint motion over entire video. Due to this, it can much reduce the computing cost in the classify phase. Therefore, the final feature vector is 2*15*8 dimension which 2 for both azimuth and elevation, 15 for joints in the skeleton, 8 for the sparse coefficients of dictionary. Fig. 7 shows the concept of sparse representation in our system.

4. EXPERIMENT

In this chapter, we will illustrate the detail of our experiment which includes the dataset, parameter and tools.

4.1 Dataset: MSR Action3D Dataset

It is a public dataset proposed in [27]. In this dataset, there are 20 action types, including 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. While doing the action, each subject were directly facing the camera. Each action will be performed by 10 different subjects, and each subject performs each action 2 or 3 times. Totally, there are 567 depth map sequences in this dataset. To explain more, the resolution is 320x240 and the data was recorded with a depth sensor similar to the Kinect device under 15 frame rate. In addition, if the action were performed by single limb like arm or leg, the subjects would use their right arm or leg.

4.2 Implement and Experimental Result

Fig. 9 shows the entire process of our recognition system. In the experiment, we will normalize all the video clips to 50 frames before training or testing that we can assure the feature vector dimension of each individual one is the same. We equally divide original data into 50 new sample points and the values of sample points which locate at noninteger position will be derived from the weighted sum of two nearest neighbor that is well-known as Interpolation. Fig. 8 shows an example of our normalization process which transform a video clips with 40 frames into 50 frames. Moreover, we set the size of the dictionary to 8 and the dimension of feature vector is therefore 240 as we mentioned in chapter 3.2. Also, we use Support Vector Machine [28] with linear kernel as our classifier for efficiency. It only takes about 0.17 second to classify each testing video with an I5 processor computer that shows the efficiency of our approach. Table 1 illustrates the detail of our experimental environment. For comparison of classifier, we also apply HMM as classifier. However, it results in worse performance and takes much more time. To evaluate our system, we compare the classification accuracy of our final representation with raw image and the basic relative angle descriptor in Section 3.1, and both SVM and HMM are used as classifier. Table 2 shows the result while using leave-one-out cross validation framework, that is, we will compute the classification accuracy of all the subjects in the dataset one by one and calculate their average value as the final result .

Being the same as [27] and [26], we also divide all the action class in MSR Action3D Dataset into three different subset and evaluate their performance individually. Table 3 shows the action type in each subset and Table 4 shows the result of the classification accuracy. We apply three also different testing scenarios. In the test one scenario, one third of the training samples were used as training data and the remaining were therefore for testing. In the test two, two third of the samples were for training and one third for testing. In cross subject test, both testing and training data comprise half of the samples in the dataset. We conducted experiment for all combination of the testing scenarios and action subset and compute their overall performance.

5. CONCLUSION

We propose a novel and easy-implementing representation for action recognition which can generally describe the motion of human skeleton in this paper. The sparse structure and low requirement of dimension result in its computing efficiency. Moreover, we construct a spherical coordinate system with respect to human skeleton itself and use relative angle between each successive joint pair as our feature that this representation can be highly tolerant of scale, viewpoint, appearance and lighting variation.

In the future work, we want to apply Dynamic Time Warping (DTW) to replace the step of linear normalization in our current system. Since the speed of doing an action may seriously vary for each individual subject, linear normalization will result in bad performance. On the other hand, DTW will automatically find out the best alignment of two time series which can much greatly deal with temporal variation of human action.

Table 1: Environment setting.

Component	
CPU	Intel I5 dual cores
Memory	4GB
Operating System	Windows 8
Coding Platform	Matlab

Table 2: Result of MSR Action3D Dataset.

Features and classifier	*Accuracy*
Raw joint position + linear SVM	67.0965
Relative Angle of Skeletal Tree + Linear SVM	68.43
Sparse representation of skeletal tree + HMM	62.08.
Sparse representation of skeletal tree + linear SVM	72.66

Table 3: The three different subset.

Action Set 1	*Action Set 2*	*Action Set 3*
Horizontal arm wave	High arm wave	High throw
Hammer	Hand catch	Forward kick
Forward punch	Draw x	Side kick
High throw	Draw tick	Jogging
Hand clap	Draw circle	Tennis swing
Bend	Two hand wave	Tennis serve
Tennis serve	Forward kick	Golf swing
Pickup & throw	Side boxing	Pickup&throw

Table 4: Result of different subset in three testing case.

	Test 1	*Test 2*	*Cross Subject Test*
AS1	83.73	87.99	83.74
AS2	84.63	87.79	84.03
AS3	88.34	93.78	92.05
Overall	85.56	89.85	86.61

REFERENCES

[1] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, *et al.*, "Real-time human pose recognition in parts from single depth images," *Communications of the ACM,* vol. 56, pp. 116-124, 2013.

[2] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision,* vol. 60, pp. 91-110, 2004.

[3] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European Conference on Computer Vision*, ed: Springer, 2006, pp. 404-417.

[4] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005, pp. 886-893.

[5] J. Perš, V. Sulić, M. Kristan, M. Perše, K. Polanec, and S. Kovačič, "Histograms of optical flow for efficient representation of body motion," *Pattern Recognition Letters,* vol. 31, pp. 1369-1376, 2010.

[6] L. Yeffet and L. Wolf, "Local trinary patterns for human action recognition," in *IEEE International Conference on Computer Vision*, 2009, pp. 492-497.

[7] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *International journal of computer vision,* vol. 103, pp. 60-79, 2013.

[8] A. Ciptadi, M. S. Goodwin, and J. M. Rehg, "Movement Pattern Histogram for Action Recognition and Retrieval," *European Conference on Computer Vision,* 2014.

[9] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *IEEE International Conference on Computer Vision* 2013, pp. 3551-3558.

[10] A. Klaser, M. Marszałek, and C. Schmid, "A spatio-temporal descriptor based on 3d-gradients," in *British Machine Vision Conference*, 2008, pp. 275: 1-10.

[11] M. B. Holte, B. Chakraborty, J. Gonzalez, and T. B. Moeslund, "A local 3-D motion descriptor for multi-view human action recognition from 4-D spatio-temporal interest points," *IEEE Journal of Selected Topics in Signal Processing,* vol. 6, pp. 553-565, 2012.

[12] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," in *International Conference on Multimedia*, 2007, pp. 357-360.

[13] 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*, 2013, pp. 716-723.

[14] X. Wu, D. Xu, L. Duan, and J. Luo, "Action recognition using context and appearance distribution features," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 489-496.

[15] C. Lu, J. Jia, and C.-K. Tang, "Range-sample depth feature for action recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 772-779.

[16] M. Zhang and A. A. Sawchuk, "Motion primitive-based human activity recognition using a bag-of-features approach," in *ACM SIGHIT International Health Informatics Symposium*, 2012, pp. 631-640.

[17] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Annual International Conference on Machine Learning*, 2009, pp. 689-696.

[18] M. Z. Uddin, N. D. Thang, J. T. Kim, and T.-S. Kim, "Human activity recognition using body joint-angle features and hidden Markov model," *Etri Journal,* vol. 33, pp. 569-579, 2011.

[19] E. Ohn-Bar and M. M. Trivedi, "Joint angles similarities and HOG2 for action recognition," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2013, pp. 465-470.

[20] 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," *Journal of Visual Communication and Image Representation,* vol. 25, pp. 24-38, 2014.

[21] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human action recognition by representing 3d skeletons as points in a lie group," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 588-595.

[22] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng, "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 3361-3368.

[23] L. Sun, K. Jia, T.-H. Chan, Y. Fang, G. Wang, and S. Yan, "DL-SFA: Deeply-Learned Slow Feature Analysis for Action Recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2625-2632.

[24] L. R. Rabiner and B.-H. Juang, "An introduction to hidden Markov models," *IEEE ASSP Magazine,* vol. 3, pp. 4-16, 1986.

[25] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *The Journal of Machine Learning Research,* vol. 10, pp. 207-244, 2009.

[26] L. Xia, C.-C. Chen, and J. Aggarwal, "View invariant human action recognition using histograms of 3d joints," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2012, pp. 20-27.

[27] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3d points," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2010, pp. 9-14.

[28] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST),* vol. 2, p. 27, 2011.