EigenJoints-based Action Recognition Using Naïve-Bayes-Nearest-Neighbor

Xiaodong Yang and YingLi Tian
Department of Electrical Engineering
The City College of New York, CUNY
{xyang02, ytian}@ccny.cuny.edu

Abstract

In this paper, we propose an effective method to recognize human actions from 3D positions of body joints. With the release of RGBD sensors and associated SDK, human body joints can be extracted in real time with reasonable accuracy. In our method, we propose a new type of features based on position differences of joints, EigenJoints, which combine action information including static posture, motion, and offset. We further employ the Naïve-Bayes-Nearest-Neighbor (NBNN) classifier for multi-class action classification. The recognition results on the Microsoft Research (MSR) Action3D dataset demonstrate that our approach significantly outperforms the state-of-the-art methods. In addition, we investigate how many frames are necessary for our method to recognize actions on the MSR Action3D dataset. We observe 15-20 frames are sufficient to achieve comparable results to that using the entire video sequences.

1. Introduction

Automatic human action recognition has been widely applied in a number of real-world applications, e.g. video surveillance, content-based video search, human-computer interaction, and health-care. Traditional research mainly concentrates on action recognition from video sequences captured by a single camera. In this case, a video is a sequence of 2D frames with RGB channels in chronological order. There has been extensive research in the literature on action recognition for such videos. The spatio-temporal volume-based method is extensively used by measuring the similarity between two action volumes. In order to compute accurate similarity measurement, a variety of spatio-temporal volume detection and representation approaches have been proposed [2, 4-7]. Trajectory-based methods have also been widely explored for recognizing human activities [11, 14]. In these methods, human actions can be interpreted by a set of body joints or other interesting points. However, it is not trivial to quickly and reliably extract and track body joints from traditional 2D videos. On the other hand, as the imaging technique advances, especially the launch of Microsoft Kinect, it has become practical to capture RGB sequences as well as depth maps in real time. Depth maps are able to provide additional body shape information to differentiate actions that have similar 2D projections from a single view. Li *et al.* [8] sampled 3D representative points from the contours of depth maps of a body surface projected onto three orthogonal Cartesian planes. An action graph was then used to model the sampled 3D points for recognition. Their promising recognition results on the MSR Action3D dataset [15] validated the superiority of 3D silhouettes over 2D silhouettes that are from a single view. However, in their experiments depth maps incurred a great amount of data which resulted in prohibitively expensive computations in clustering training samples of all classes.

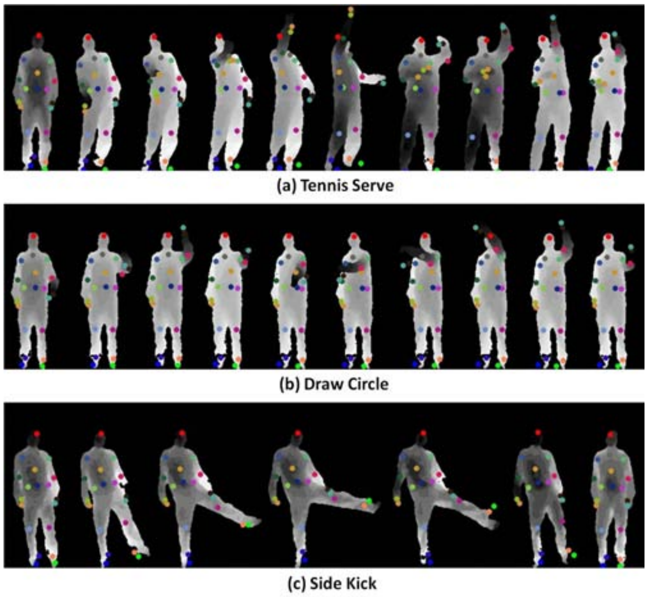

Figure 1: The sequences of depth maps and body joints for actions of (a) Tennis Serve, (b) Draw Circle, and (c) Side Kick. Each depth map includes 20 joints. The joint of each body part is encoded in corresponding color.

The biological observation from Johansson [9] suggested that human actions could be modeled by the motion of a set of key joints. With the release of RGBD sensors and the associated SDK, we are able to obtain 3D positions of body joints in real time with reasonable accuracy [13]. In this paper, we focus on recognizing human actions using body joints extracted from sequences of depth maps. Fig. 1 demonstrates the depth sequences with 20 extracted body joints of each depth map for actions *Tennis Serve*, *Draw Circle*, and *Side Kick*. As shown in Fig. 1, the perception of actions can be reflected by the motions of individual joints and the configuration of different joints (i.e. static postures). Compared to the original depth data of human body, these joints are more compact and more distinctive. We propose a type of features by adopting the differences of joints in both temporal and spatial domains to explicitly model the dynamics of individual joints and the configuration of different joints. We then apply Principal Component Analysis (PCA) to joint differences to obtain EigenJoints by reducing redundancy and noise. We employ non-parametric Naïve-Bayes-Nearest-Neighbor (NBNN) [3] as a classifier to recognize multiple action categories. In accordance with the principles behind NBNN-based image classification, we avoid quantization of frame descriptors and compute *Video-to-Class* distance, instead of *Video-to-Video* distance. In addition, most existing methods [2, 4-7, 11-14] perform action recognition by operating on entire video sequences. We further explore how many frames are sufficient for action recognition in our framework. The experimental results on the MSR Action3D dataset show that a short sub-sequence (15-20 frames) is sufficient to perform action recognition, with quite limited gains as more frames are added in. This observation is important for making online decisions and reducing observational latency when humans interact with computers.

The remainder of this paper is organized as follows. Section 2 reviews existing methods for action recognition. In Section 3, we provide detailed procedures of extracting EigenJoints features for each frame. Section 4 describes the NBNN classifier. A variety of experimental results and discussions are presented in Section 5. Finally, Section 6 summarizes the remarks of this paper.

2. Related Work

In traditional 2D videos captured by a single camera, action recognition mainly focuses on analyzing spatio-temporal volumes. The core of these approaches is in the detection and representation of space-time volumes. For example, Bobick and Davis [2] stacked foreground regions of a person to explicitly track shape changes. The stacked silhouettes formed Motion History Images (MHI) and Motion Energy Images (MEI), which served as action descriptors for template matching. In most recent work, local spatio-temporal features have been widely used. Similar to object recognition using sparse local features in 2D images, an action recognition system first detects interesting points (e.g. STIPs [6] and Cuboids [4]) and then computes descriptors (e.g. HOG/HOF [7] and HOG3D [5]) based on the detected local motion volumes. These local features are then combined (e.g. bag-of-words) to model different activities. The trajectory-based approaches are more similar to our method that models actions by the motion of a set of points. For instance, Rao and Shah [11] used skin color detection to track a hand position to record its 3D (XYT) space-time trajectory curve. They represented actions by a set of peaks of trajectory curves and intervals between the peaks. Sun *et al.* [14] extracted trajectories through pair-wise SIFT matching between neighboring frames. The stationary distribution of a Markov chain model was then used to compute a velocity description.

As RGBD sensors becomes available, research of action recognition based on depth information has been explored. Li *et al.* [8] proposed a Bag-of-3D-Points model for action recognition. They sampled a set of 3D points from a body surface to characterize the posture being performed in each frame. In order to select the representative 3D points, they first sampled 2D points at equal distance along the contours of projections formed by mapping the depth map onto three orthogonal Cartesian planes, i.e. XY, XZ, and YZ planes. The 3D points were then retrieved in the 3D depth map. Their experiments showed that this approach considerably outperformed the methods only using 2D silhouette and were more robust to occlusion.

Motivated by the robust joints extraction of RGBD sensors and associated SDK, we propose to compute EigenJoints for action recognition. In contrast to traditional trajectory-based methods, EigenJoints are able to model actions by more informative and more accurate body joints without background noisy points. Compared to the 3D silhouette based recognition, EigenJoints are more discriminative and much more compact.

3. Representation of EigenJoints

The proposed framework to compute EigenJoints is demonstrated in Fig. 2. We employ 3D position differences of joints to characterize action information including posture feature f_{cc}, motion feature f_{cp}, and offset feature f_{ci} in each frame-c. We then concatenate the three features channels as $f_c = [f_{cc}, f_{cp}, f_{ci}]$. According to different experimental settings in Section 5.1, two normalization schemes are introduced to obtain f_{norm}. In the end, PCA is applied to f_{norm} to compute EigenJoints.

As shown in Fig. 2, the 3D coordinates of N joints are available in each frame: $X = \{x_1, x_2, \ldots, x_N\}$. To characterize the static posture information of current frame-c, we compute pair-wise joints differences within the current frame:

$$f_{cc} = \{x_i - x_j \mid i, j = 1,2, \ldots, N; i \neq j\} \qquad (1)$$

To capture the motion property of current frame-c, the pair-wise joints differences are computed between the current frame-c and the preceding frame-p:

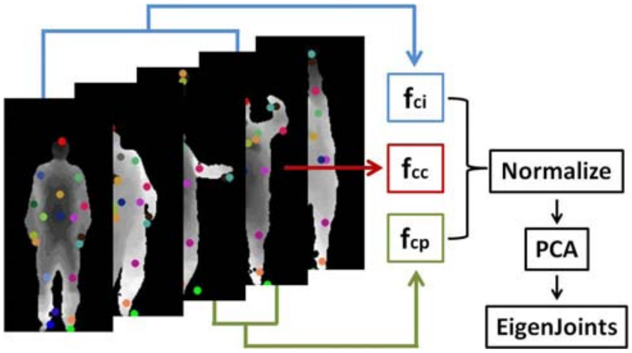

Figure 2: The framework of representing EigenJoints. In each frame, we compute three feature channels f_{ci}, f_{cc}, and f_{cp} to capture the information of offset, posture, and motion. The normalization and PCA are then applied to obtain EigenJoints descriptor for each frame.

$$f_{cp} = \{x_i^c - x_j^p | x_i^c \in X_c; x_j^p \in X_p\} \tag{2}$$

To represent the offset feature or the overall dynamics of the current frame-c with respect to the initial frame-i, we calculate the pair-wise joints differences between frame-c and frame-i:

$$f_{ci} = \{x_i^c - x_j^i | x_i^c \in X_c; x_j^i \in X_i\} \tag{3}$$

The initial frame tends to approximate the neutral posture. The combination of the three feature channels forms the preliminary feature representation for each frame: $f_c = [f_{cc}, f_{cp}, f_{ci}]$.

However, the 3 attributes (u, v, d) of a joint x might be of inconsistent coordinates, e.g. (u, v) are screen coordinates and d is world coordinate. So the normalization is then applied to f_c to avoid attributes in greater numeric ranges dominating those in smaller numeric ranges. We use linear normalization scheme to scale each attribute in f_c to the range $[-1, +1]$. The other advantage of normalization is to reduce intra-class variations under different test sets. So we normalize f_c based on a single video for cross-subject test and based on entire training videos for non-cross-subject test.

As shown in Fig. 1, in each frame we use $N = 20$ joints which result in a huge feature dimension. f_{cc}, f_{cp}, and f_{ci} contains 190, 400, and 400 pair-wise comparisons, respectively. Each comparison generates 3 attributes $(\Delta u, \Delta v, \Delta d)$. In the end, f_{norm} is with the dimension of $(190 + 400 + 400) \times 3 = 2970$. We employ PCA to reduce redundancy and noise in f_{norm}, and to obtain the compact EigenJoints representation for each frame. In the experimental results of Section 5.2, we observe that most energy is covered in a first few leading eigenvectors, e.g. the first 128 eigenvalues weight over 95%.

4. Naïve-Bayes-Nearest-Neighbor Classifier

We employ the Naïve-Bayes-Nearest-Neighbor (NBNN) [3] as the classifier for action recognition. The Nearest-Neighbor (NN) is a non-parametric classifier which has some advantages over most learning-based classifiers: (1) naturally deal with a large number of classes; (2) avoid the overfitting problem; (3) require no learning process. Boiman *et al.* [3] argued that the effectiveness of NN was largely undervalued by the quantization of local image descriptors and the computation of *Image-to-Image* distance. Their experiments showed that frequent descriptors had low quantization error but rare descriptors had high quantization error. However, discriminative descriptors tend to be rare. So quantization significantly degrades the discriminative power of descriptors. In addition, they observed that *Image-to-Class* that made use of the descriptor distribution over an entire class provided better generalization capacity than *Image-to-Image*.

We extend these concepts of NBNN-based image classification to NBNN-based video classification (action recognition). We directly use the frame descriptors of EigenJoints without quantization, and compute *Video-to-Class* distance rather than *Video-to-Video* distance. In the context of NBNN, the action recognition is performed by:

$$C^* = \arg\min_c \sum_{i=1}^{M} \|d_i - NN_c(d_i)\|^2 \tag{4}$$

where $d_i, i = 1,2, \dots, M$ is an EigenJoints descriptor of frame-i in a testing video; M is the number of frames; $NN_c(d_i)$ is the nearest neighbor of d_i in class-C. The experimental results in Section 5.3 show that the recognition accuracy based on NBNN outperforms that based on SVM. The efficient approximate-r-nearest-neighbours algorithm and KD-tree [1] can be used to reduce the computational cost in NBNN classification.

5. Experiments and Discussions

We evaluate our proposed method on the MSR Action3D dataset [8, 15]. We extensively compare the state-of-the-art methods to our approach under different experimental settings. In addition, we investigate how many frames in a testing video are sufficient to perform action recognition in our framework.

5.1. Experimental Setup

The MSR Action3D [15] is a public dataset that provides sequences of depth maps and skeletons captured by a RGBD camera. It includes 20 actions performed by

10 subjects facing the camera during performance. Each subject performed each action 2 or 3 times. The depth maps are with the resolution of 320×240. For each skeleton joint, the horizontal and vertical positions are stored in screen coordinates, and depth value is stored in world coordinate. The 20 actions are chosen in the context of interactions with game consoles. As shown in Fig. 1, actions in this dataset reasonably capture a variety of motions related to arms, legs, torso, and their combinations.

In order to facilitate a fair comparison, we follow the same experimental settings as [8] to split 20 actions into three subsets as listed in Table 1. In each subset, there are three different tests: Test One (One), Test Two (Two), and Cross Subject Test (CrSub). In Test One, 1/3 of the subset is used as training and the rest as testing; in Test Two, 2/3 of the subset is used as training and the rest as testing. Both of them are non-cross-subject tests. In Cross Subject Test, 1/2 of subjects are used as training and the rest ones used as testing.

Table 1: The three action subsets used in our experiments.

Action Set 1 (AS1)	Action Set 2 (AS2)	Action Set 3 (AS3)
Horizontal Wave(HoW)	High Wave(HiW)	High Throw(HT)
Hammer(H)	Hand Catch(HC)	Forward Kick(FK)
Forward Punch(FP)	Draw X(DX)	Side Kick(SK)
High Throw(HT)	Draw Tick(DT)	Jogging(J)
Hand Clap(HC)	Draw Circle(DC)	Tennis Swing(TSw)
Bend(B)	Hands Wave(HW)	Tennis Serve(TSr)
Tennis Serve(TSr)	Forward Kick(FK)	Golf Swing(GS)
Pickup Throw(PT)	Side Boxing(SB)	Pickup Throw(PT)

5.2. Evaluations of EigenJoints and NBNN

We first evaluate the energy distributions of joints differences to determine the dimensionality of EigenJoints. Fig. 3 shows the ratios between the sum of first few eigenvalues and the sum of all eigenvalues of f_{norm} under different test sets. As shown in this figure, the first 128 eigenvalues (out of 2970) occupy over 95% energy for all experimental settings. The distributions concentrate more in the first few leading eigenvalues for Test One and Test Two, where the first 32 eigenvalues have already weighted over 95%. The distribution scatters relatively more for Cross Subject Test, where the leading 32 eigenvalues cover about 85% of overall energy.

Fig. 4 demonstrates the action recognition rates of EigenJoints-based NBNN with different dimensions under different test sets. It's interesting to observe that the overall recognition rates of various test sets are very similar across different dimensions. As for each dimensionality, our method performs very well for Test One and Test Two which are non-cross-subject tests. While the performance in AS3CrSub is promising, the accuracies in AS1CrSub and AS2CrSub are relatively low. This is probably because actions in AS1 and AS2 are with similar movements, but AS3 groups complex but distinct actions. For example, in AS1 *Hammer* tends to be confused with *Forward Punch* and *Pickup Throw* consists of *Bend* and *High Throw*. Furthermore, in Cross Subject Test, different subjects perform actions with considerable variations but the number of subjects is small. For example, some subjects perform action of *Pickup Throw* using only one hand whereas others using two hands, which result in great intra-class variations. The cross subject performance can be improved by adding more subjects.

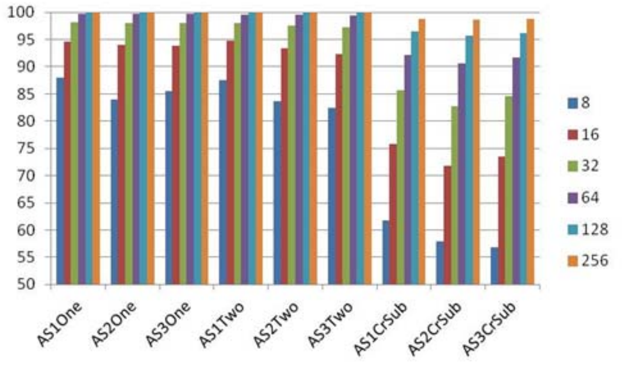

Figure 3: The ratios (%) between the sum of the first few (8, 16, 32, 64, 128, and 256) leading eigenvalues and the sum of all eigenvalues of f_{norm} under different test sets.

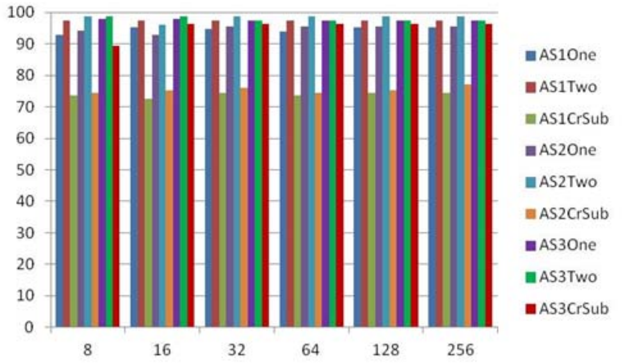

Figure 4: The recognition rates (%) of NBNN-based EigenJoints with different dimensionalities under various test sets.

Considering the recognition accuracy and the computational cost in NBNN classification, we choose 32 as the dimensionality for EigenJoints in all of our experiments. As high accuracies of Test One and Test Two (over 95%, see Table 2), we only show the confusion matrix of our method under Cross Subject Test in Fig. 5. Because of the considerable variations in actions performed by different subjects, cross subjects generate much larger intra-class variance than non-cross subjects. In AS1CrSub, most actions are confused with *Pickup Throw*, especially for *Bend* and *High Throw*. In AS2CrSub, *Draw X*, *Draw Tick*, and *Draw Circle* are

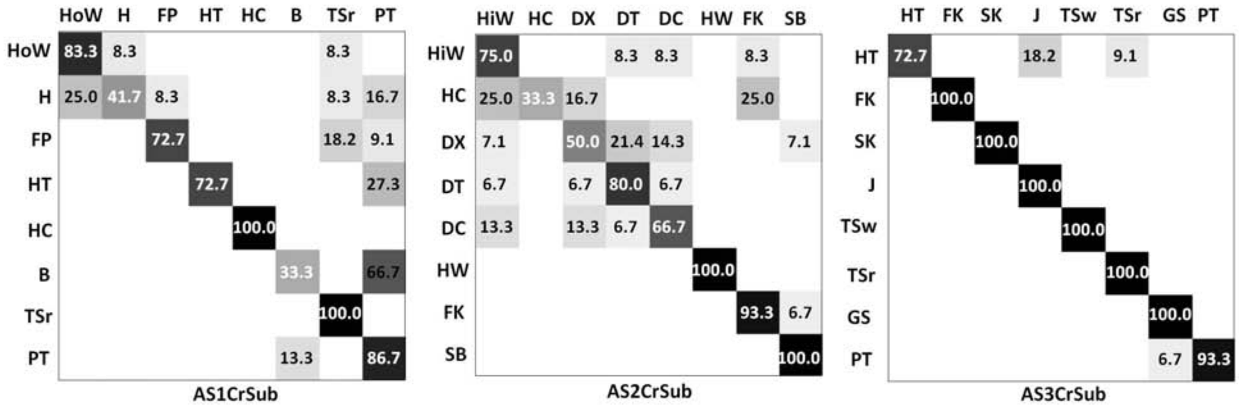

Figure 5: Confusion matrix of EigenJoints-based NBNN in different action sets of Cross Subject Test. Each row corresponds to ground truth label and each column denotes the recognition results.

mutually confused, as they contain highly similar movements. As actions in AS3 are with significant differences, the recognition results are greatly improved in AS3CrSub.

5.3. Comparison with State-of-the-art

SVM has been extensively used in computer vision to achieve the state-of-the-art performances in image and video classifications. We employ bag-of-words to represent an action video by quantizing EigenJoints of each frame. K-means clustering is employed to build the codebook. We empirically choose K = 100 and RBF kernels to perform classification. The optimal parameters of RBF kernels are obtained by 5-fold cross-validation. Fig. 6 compares the recognition results based on NBNN and SVM. As shown in this figure, NBNN consistently outperforms SVM in all test sets. This result validates the superiority of the two schemes used in NBNN, i.e. non-quantization of EigenJoints and *Video-to-Class* distance.

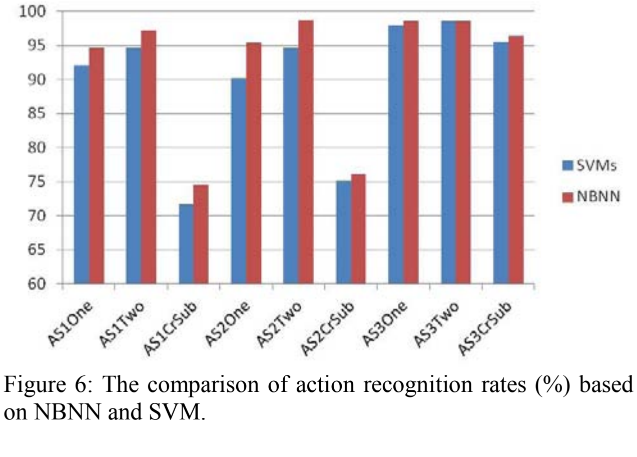

Figure 6: The comparison of action recognition rates (%) based on NBNN and SVM.

We further compare our approach with the state-of-the-art method [8] for action recognition on the MSR Action3D dataset [15] in Table 2. The results of Bag-of-3D-Points or 3D silhouettes are obtained from paper [8]. The best recognition rates of various test sets are highlighted in bold. As shown in Table 2, our method consistently and significantly outperforms 3D silhouettes. For example, the average accuracies of our method in Test One, Test Two, and Cross Subject Test are 95.8%, 97.8%, and 81.4%, which outperform the average accuracies of using 3D silhouettes [8] by 4.2%, 3.6%, and 6.7% respectively. In non-cross subject tests, our method achieves over 95% accuracies in most cases. While the accuracy of AS3CrSub is 96.4%, the recognition rates of cross subject tests in AS1CrSub (74.5%) and AS2CrSub (76.1%) are relatively low. This is probably because similar actions in AS1CrSub and AS2CrSub are more sensitive to the larger intra-class variations generated in cross subject tests. In addition to recognition accuracy, our method is much more compact than the Bag-of-3D-Points.

Table 2: Recognition rates (%) of our method compared to the state-of-the-art approach on the MSR Action3D dataset.

	3D Silhouettes [8]	our method
AS1One	89.5	**94.7**
AS2One	89.0	**95.4**
AS3One	96.3	**97.3**
AS1Two	93.4	**97.3**
AS2Two	92.9	**98.7**
AS3Two	96.3	**97.3**
AS1CrSub	72.9	**74.5**
AS2CrSub	71.9	**76.1**
AS3CrSub	79.2	**96.4**

5.4. How Many Frames Are Sufficient

Li *et al*. [8] recognized actions using entire video sequences (about 50 frames) in the MSR Action3D dataset. We perform experiments to investigate how many

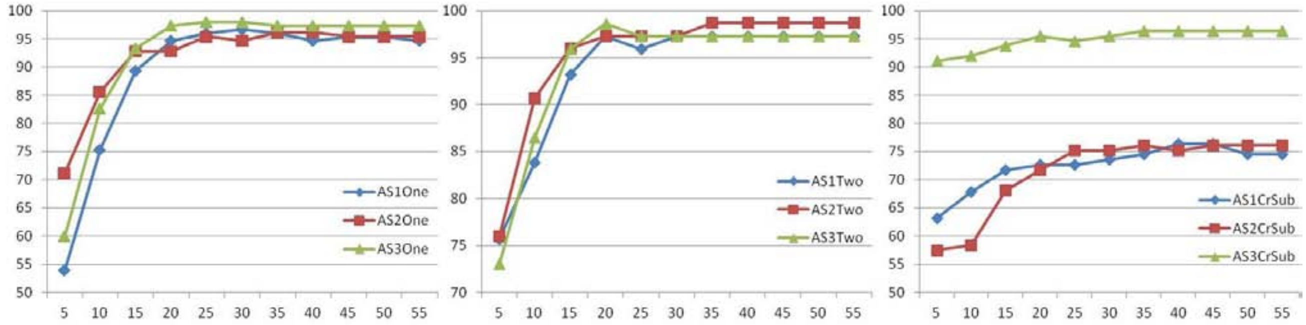

Figure 7: The recognition accuracies using different number of frames in Test One (left), Test Two (middle), and Cross Subject Test (right). 15-20 frames are sufficient to enable action recognition in most test sets.

frames are sufficient for action recognition in our framework. The recognition accuracies using different number of frames under a variety of test sets are given in Fig. 7. The sub-sequences are extracted from the first T frames of a given video. As shown in this figure, in most cases 15-20 frames are sufficient to achieve comparable recognition accuracies to the ones using entire sequences. There are rapid diminishing gains as more frames are added in. This result is also in accordance with the observations in [10] that a 66% reduction in frames only results in a 6.6% reduction in classification accuracy. These results are highly relevant for action recognition systems where decisions have to be made on line.

6. Conclusion

In this paper, we have proposed an EigenJoints-based action recognition system using an NBNN classifier. The compact and discriminative frame representation of EigenJoints is able to capture the properties of posture, motion, and offset of each frame. The comparisons between NBNN and SVM show that non-quantization of descriptors and *Video-to-Class* distance computation are more effective for action recognition. The experimental results on the MSR Action3D dataset demonstrate our approach significantly outperforms the state-of-the-art method based on 3D silhouettes. In addition, we observe that 15-20 frames are sufficient to perform action recognition with reasonably accurate results. Future work will focus on incorporating more subjects to improve recognition in the cross subject test.

Acknowledgement

This work was supported in part by PSC-CUNY # 64720-00 42 and Microsoft Research.

References

[1] S. Arya and H. Fu. Expected-Case Complexity of Approximate Nearest Neighbor Searching. In *Symposium of Discrete Algorithms*, 2000.

[2] A. Bobick and J. Davis. The Recognition of Human Movement Using Temporal Templates. *IEEE Trans. on PAMI*, 2001.

[3] O. Boiman, E. Shechtman, and M. Irani. In Defense of Nearest-Neighbor Based Image Classification. In *CVPR*, 2008.

[4] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior Recognition via Sparse Spatio-Temporal Features. In *VS-PETS*, 2005.

[5] A. Klaser, M. Marszalek, and C. Schmid. A Spatio-Temporal Descriptor based on 3D Gradients. In *BMVC*, 2008.

[6] I. Laptev. On Space-Time Interest Points. *International Journal of Computer Vision*, 2005.

[7] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning Realistic Human Actions from Movies. In *CVPR*, 2008.

[8] W. Li, Z. Zhang, and Z. Liu. Action Recognition based on A Bag of 3D Points. *IEEE Workshop on CVPR for Human Communicative Behavior Analysis*, 2010.

[9] G. Johansson. Visual Perception of Biological Motion and A Model for Its Analysis. *Perception and Psychophysics*, 1973.

[10] S. Masood, C. Ellis, M. Tappen, J. Laviola, and R. Sukthankar. Measuring and Reducing Observational Latency When Recognizing Actions. *IEEE Workshop on ICCV for Human Computer Interaction: Real Time Vision Aspects of Natural User Interfaces*, 2011.

[11] C. Rao and M. Shah. View-Invariance in Action Recognition. In *CVPR*, 2001.

[12] K. Schindler and L. Gool. Action Snippets: How Many Frames Does Human Action Recognition Require? In *CVPR*, 2008.

[13] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-Time Pose Recognition in Parts from Single Depth Images. In *CVPR*, 2011.

[14] J. Sun, X. Wu, S. Yan, L. Cheong, T. Chua, and J. Li. Hierarchical Spatio-Temporal Context Modeling for Action Recognition. In *CVPR*, 2009.

[15] http://research.microsoft.com/en-us/um/people/zliu/ActionRecoRsrc/default.htm