Effective 3D Action Recognition Using EigenJoints

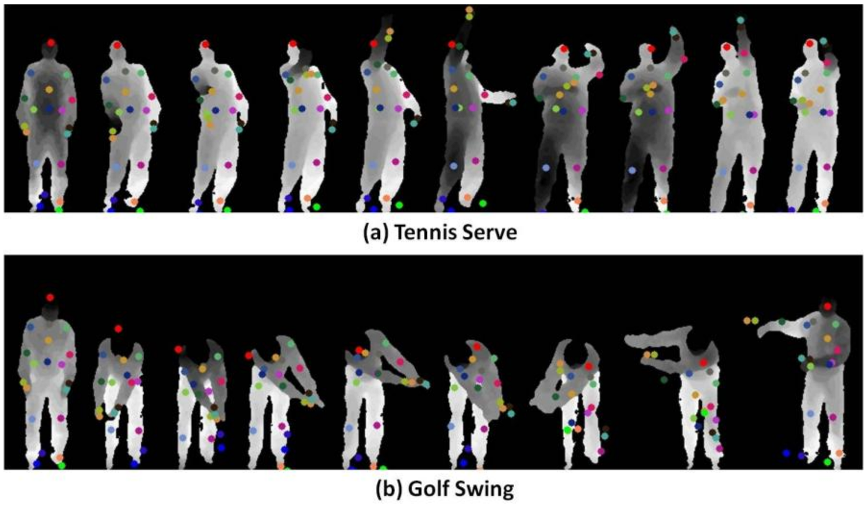
Xiaodong Yang and YingLi Tian
Department of Electrical Engineering
The City College of New York, CUNY
{xyang02, ytian}@ccny.cuny.edu

Abstract — In this paper, we propose an effective method to recognize human actions using 3D skeleton joints recovered from 3D depth data of RGBD cameras. We design a new action feature descriptor for action recognition based on differences of skeleton joints, i.e., EigenJoints which combine action information including static posture, motion property, and overall dynamics. Accumulated Motion Energy (AME) is then proposed to perform informative frame selection, which is able to remove noisy frames and reduce computational cost. We employ non-parametric Naïve-Bayes-Nearest-Neighbor (NBNN) to classify multiple actions. The experimental results on several challenging datasets demonstrate that our approach outperforms the state-of-the-art methods. In addition, we investigate how many frames are necessary for our method to perform classification in the scenario of online action recognition. We observe that the first 30% to 40% frames are sufficient to achieve comparable results to that using the entire video sequences on the MSR Action3D dataset.

Keywords — action recognition, RGBD camera, depth data, skeleton joints, 3D action feature representation

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 [5, 16, 18, 27, 29, 31]. Traditional research mainly concentrates on action recognition of video sequences captured by RGB cameras [2, 3, 6, 9, 10, 12, 30]. In this case, a video is a sequence of 2D frames with RGB images in chronological order. There has been extensive research in the literature on action recognition for such 2D videos. The spatio-temporal volume-based methods have been extensively used by measuring the similarity between two action volumes. In order to enable accurate similarity measurement, a variety of spatio-temporal volume detection and representation methods have been proposed [3, 6, 9, 10, 27]. Trajectory-based approaches have been widely explored for recognizing human activities as well [5, 12, 22]. In these methods, human actions can be interpreted by a set of key joints or other interesting points. However, it is not trivial to quickly and reliably extract and track skeleton joints from traditional RGB videos. On the other hand, as imaging techniques advance, such as RGBD cameras of Microsoft Kinect and ASUS Xtion Pro Live, 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. It has therefore motivated recent research work to investigate action recognition using the 3D information. Li *et al.* [11] 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 experimental results validated the superiority of 3D silhouettes over 2D silhouettes from a single view. Xia *et al.* [25] mapped 3D joints to a spherical coordinate system and used histogram of modified 3D joint positions to achieve view-invariant posture representation. Sung *et al.* [23] made use of features extracted from RGB images, depth maps, and skeleton joints to recognize human activities in multiple indoor environments. Ellis *et al.* [14] employed a latency aware learning method for action recognition and studied the trade-off between recognition accuracy and observational latency.



(a) Tennis Serve

(b) Golf Swing

Figure 1: Sampled sequences of depth maps and skeleton joints in actions of (a) *Tennis Serve* and (b) *Golf Swing*. Each depth map includes 20 joints. The joints of each body part are encoded in corresponding colors.

The biological observation from Johansson [13] suggested that human actions could be modeled by the motion of a set of skeleton joints. The MoCap system [17] was used to extract 3D joint positions by using markers and high precision camera array. With the release of RGBD cameras and the associated SDK, we are able to recover 3D positions of skeleton joints in real time and with reasonable accuracy [7, 20, 21]. In this paper, we focus on recognizing human actions using skeleton joints extracted from sequences of depth maps. Fig. 1 demonstrates the depth sequences with 20 extracted skeleton joints in each depth map of actions *Tennis Serve* and *Golf Swing*. As illustrated in this figure, the perception of each action can be reflected by the motions of individual joints (i.e., motion property) and the configuration of different joints (i.e., static postures). Compared to point cloud of human body in depth maps, these skeleton joints are much more compact.

In this paper, we design a novel action feature descriptor by adopting the differences of skeleton joints in both temporal and spatial domains to explicitly model the dynamics of each individual joint and the configuration of different joints. We then apply Principal Component Analysis (PCA) to the joint differences to obtain EigenJoints by reducing redundancy and noise. Similar to the affect recognition in [8], the temporal segments of an action can be intuitively approximated by the statuses of neutral, onset, apex, and offset. The discriminative information is however not evenly distributed in the four statuses. We propose a measurement of Accumulated Motion Energy (AME) to quantize the distinctiveness of each frame. The less distinctive frames are then pruned to remove noise and reduce computational cost. We employ non-parametric Naïve-Bayes-Nearest-Neighbor (NBNN) [4] as the 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 perform action recognition by operating on entire video sequences. However, this is not practical to online systems which require as few frames as possible for recognition. So we further investigate how many frames are sufficient to obtain reasonably accurate action recognition in our framework. Experimental results on the MSR Action3D dataset [11, 32] demonstrate that a short sub-sequence (e.g., the first 30%-40% frames) of the entire video 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 latency when humans interact with computers.

An earlier version of this paper can be found in [26]. Compared to our previous work, there are three major extensions that merit being highlighted: 1) selection of informative frames based on Accumulated Motion Energy (AME); 2) extensive experiments on more challenging datasets including the MSR

Action3D [11, 32], the Cornell Human Activity [23, 33], and the UCF Kinect [14, 34]; and 3) more comparisons with the state-of-the-art work.

The remainder of this paper is organized as follows. Section 2 reviews existing methods for human action recognition. In Section 3, we provide detailed procedures of extracting EigenJoints from each frame. Section 4 briefly introduces NBNN classifier. Section 5 describes informative frame selection by using Accumulated Motion Energy (AME). A variety of experimental results and discussions are presented in Section 6. Finally, Section 7 summarizes the remarks of this paper.

2. Related Work

In traditional RGB videos, human action recognition mainly focuses on analyzing spatio-temporal volumes. The core of these approaches is the detection and representation of space-time volumes. Bobick and Davis [3] 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 [10], Cuboids [6], and SURF+MHI [27]) and then computes descriptors (e.g., HOG/HOF [10] and HOG3D [9]) based on the detected local motion volumes. These local features are then combined (e.g., bag-of-words) to represent actions. The trajectory-based approaches are more similar to our method that model actions by the motion of a set of points of human body parts. Sun *et al.* [22] 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.

The availability of 3D sensors has recently made it possible to capture depth maps in real time, which has facilitated a variety of visual recognition tasks, such as human pose estimation and human action recognition. Shotton *et al*. [20] proposed an object recognition method to predict 3D positions of body joints from a single depth image. This scheme was further extended in [7] and [21] by aggregating votes from a regression forest and incorporating dependency relationships between body part locations, respectively. With the release of RGBD cameras and associated SDK, research of action recognition based on depth information and skeleton joints has also been explored as well. Li *et al.* [11] 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 point cloud of depth maps. Their experiments on the MSR Action3D dataset [11] showed that this approach considerably outperformed the methods only using 2D silhouette and were more robust to occlusion. However, in their experiments sampling of 3D points incurred a great amount of data which resulted in expensive computations in clustering training samples of all classes. In [25] Xia *et al.* used Histogram of 3D Joint Locations (HOJ3D) to represent posture. They transferred skeleton joints into a spherical coordinate to achieve view-invariance. The temporal information was then coded by discrete Hidden Markov Models (HMM). Sung *et al.* [23] employed both visual (RGB) and depth (D) channels to recognize human daily activities. The skeleton joints were used to model body pose, hand position, and motion information. They also extracted Histogram of Oriented Gradients (HOG) features from region of interest in gray images and depth maps to characterize the appearance information. A hierarchical Maximum Entropy Markov Model (MEMM) was then used to decompose an activity to a set of sub-activities and perform action recognition on the Cornell Human Activity dataset [23]. Yang *et al.* [28] projected 3D depth maps onto three 2D orthogonal planes that were stacked as Depth Motion Maps (DMM). HOG was then computed from DMM as a global representation of human action. An actionlet mining algorithm was proposed in [24] to perform selection of skeleton joints. In addition to joint-based feature, they also made use of depth maps to characterize object shape and appearance.

Most of the above systems relied on entire video sequences (RGB or RGBD) to perform action recognition. As for an online scenario, a system is however supposed to require as few observations as possible. Schindler and Gool [19] first investigated how many frames were required to enable action classification in RGB videos. They found that short action snippets with a few frames (e.g., 1 – 7 frames) were almost as informative as the entire video. In order to reduce the observational latency that is the time a system takes to observe sufficient information for a good classification, Ellis *et al.* [14] proposed to recognize actions based upon an individual canonical pose from a sequence of postures. The canonical pose covered the information of posture, motion, and overall variance by using skeleton joints. They used a classifier based on logistic regression to minimize observational latency and classify actions on the UCF Kinect dataset [14].

Motivated by the robust extraction of skeleton joints using RGBD cameras and the associated SDK, we propose a new action feature descriptor, EigenJoints, for action recognition. In contrast to traditional trajectory-based methods, EigenJoints are able to model actions through more informative and more accurate body joints without background or noisy points. Compared to the state-of-the-art features using skeleton joints or depth maps, EigenJoints are more discriminative, more compact, and easier to compute.

3. Representation of EigenJoints

The proposed framework to compute EigenJoints is demonstrated in Fig. 2. We employ 3D position differences of skeleton joints to characterize action information including static posture feature f_{cc}, consecutive motion feature f_{cp}, and overall dynamics 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 (e.g., cross subject test or non-cross subject test), two normalization schemes are introduced to obtain f_{norm}. In the end, PCA is applied to f_{norm} to generate EigenJoints.

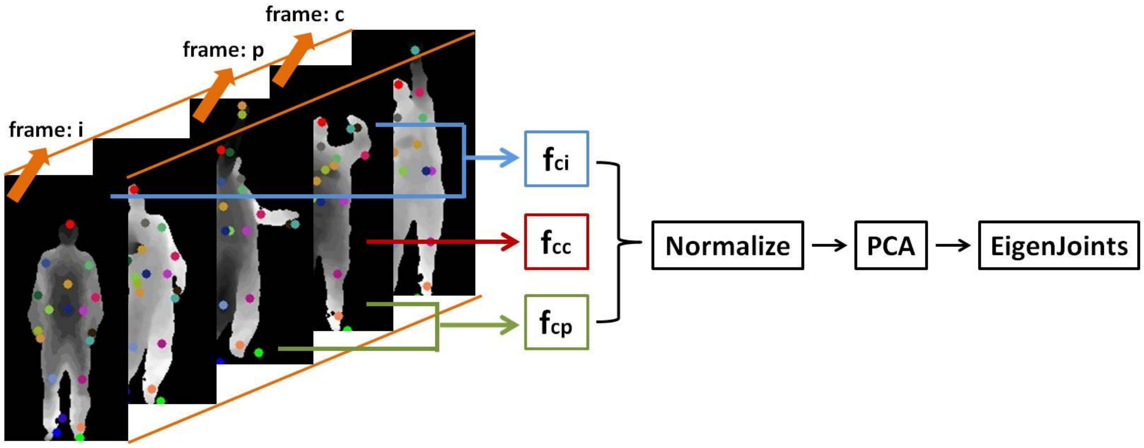


Figure 2: The framework of representing EigenJoints. In each frame, we compute three feature channels of f_{ci}, f_{cc}, and f_{cp} to capture the information of offset, posture, and motion. The normalization and PCA are then applied to obtain EigenJoints descriptor for each frame.

As shown in Fig. 2, the 3D coordinates of N joints can be obtained from human pose estimation [20] in each frame: $X = \{x_1, x_2, \dots, x_N\}$, $X \in \Re^{3\times N}$. To characterize the static posture information of current frame-c, we compute pair-wise joints differences within the current frame:

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

To capture the motion property of current frame-c, the joint differences are computed between the current frame-c and its preceding frame-p:

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

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

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

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

In Eq. (1-3) the orders of joints are in accordance to the specified joint indices. However, the three elements (u, v, d) of a joint x might be of inconsistent coordinates, e.g., (u, v) are in screen coordinates and d is in world coordinate. So normalization is then applied to f_c to avoid elements in greater numeric ranges dominating those in smaller numeric ranges. We use linear normalization scheme to scale each element in f_c to the range $[-1, +1]$. The other benefit of normalization is to reduce intra-class variations of the same action performed by different subjects. In our experiments, 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 illustrated in Fig. 1, in each frame we use N joints which might result in a huge feature dimension. f_{cc}, f_{cp}, and f_{ci} contain $N(N-1)/2$, N^2, and N^2 pair-wise comparisons, respectively. Each comparison generates 3 elements $(\Delta u, \Delta v, \Delta d)$. In the end, f_{norm} is with the dimension of $3 \times (N(N-1)/2 + N^2 + N^2)$. For example, in our method we extract 20 skeleton joints in each frame, f_{norm} is with the dimension of 2970. As skeleton joints are already high level information recovered from depth maps, this large dimension might be redundant and include noise, which can be illustrated in Fig. 5. We therefore apply PCA to reduce redundancy and noise in the centralized f_{norm}. The final compact representation is EigenJoints, which is the action descriptor of each frame. In the experimental results of Section 6.2, we observe that most eigenvalues are covered by the first few leading eigenvectors, e.g., the leading 128 eigenvalues weight over 95% on the MSR Action3D dataset.

4. Naïve-Bayes-Nearest-Neighbor Classifier

We employ the Naïve-Bayes-Nearest-Neighbor (NBNN) [4] as the classifier for action recognition. The Nearest-Neighbor (NN) is a non-parametric classifier which has several advantages over most learning-based classifiers: 1) naturally deal with a large number of classes; 2) avoid the overfitting problem; and 3) require no learning process. Boiman *et al.* [4] 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, most discriminative descriptors tend to be rare. So the quantization used in Bag-of-Words scheme significantly degrades the discriminative power of descriptors. In addition, kernel matrix used by SVM computes *Image-to-Image* distance. But they observed that the distance computation of *Image-to-Class* that made use of descriptor distributions over entire class provided better generalization than the *Image-to-Image* distance.

We follow these concepts of NBNN-based image classification to NBNN-based video classification (i.e., action recognition). We directly use 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:

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

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 experiments in Section 6.2 show that the recognition accuracy based on NBNN outperforms that based on SVM. The approximate-r-nearestneighbour algorithm, k-d tree [1], and local NBNN [15] can be used to reduce the computational cost in NBNN classification.

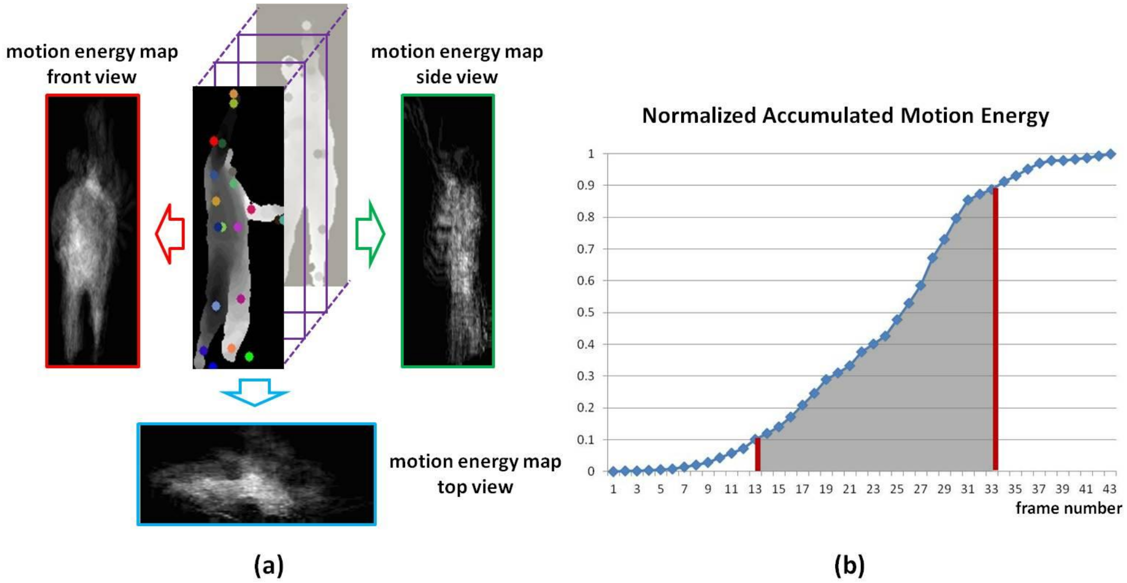


Figure 3: Computation of Accumulated Motion Energy (AME). (a) Motion energy maps associated with each projection view. (b) Normalized AME and selected informative frames.

5. Informative Frame Selection

As in the affect recognition [8], temporal segments of an action can be intuitively approximated by the statuses of neutral, onset, apex, and offset. The discriminative information is not evenly distributed in the four statuses, but concentrates more on the frames from onset and apex statuses. On the other hand, motions of neutral and offset statuses are usually similar across different action categories. So informative frame selection corresponds to extract frames from onset and apex but discard frames from neutral and offset. This process enables us to remove confusing frames and reduce computational cost in the nearest neighbor searching. We propose to use Accumulated Motion Energy (AME) to measure the distinctiveness of each frame:

$$AME(i) = \sum_{v=1}^{3} \sum_{j=1}^{i} (\left| f_v^j - f_v^{j-1} \right| > \epsilon) \qquad (5)$$

For a frame-i, its 3D depth map is first projected onto three orthogonal planes which generate three projected frames $f_v, v \in \{1,2,3\}$. $AME(i)$ is then computed as the summation of motion energy maps. The motion energy maps of each frame are obtained by thresholding and accumulating differences between two consecutive projected frames, as shown in Fig. 3(a). AME vector is then normalized by L1-norm. Fig. 3(b) illustrates a normalized AME of action *Tennis Serve* from the MSR Action3D dataset. As we can see, when normalized AME is less than 0.1 or larger than 0.9, it increases very slowly as motions in these frames are weak. It is observed that most of these frames correspond to the statuses of neutral and offset. As for the frames whose normalized AME are between 0.1 and 0.9, they present significant motions and make the curve increase dramatically. Accordingly, these frames come from the statuses of onset and apex and cover more discriminative information. In our experiment, we therefore choose frames with normalized AME between 0.1 and 0.9 as the informative frames.

6. Experiments and Discussions

We evaluate our proposed method on three challenging datasets including the MSR Action3D [11], the Cornell Human Activity [23], and the UCF Kinect [14]. We extensively compare the state-of-the-art methods to our approach under a variety of experimental settings.

6.1. Experiments on the MSR Action3D Dataset

The MSR Action3D [11] is a benchmark dataset for 3D action recognition that provides sequences of depth maps and skeleton joints 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 locations are stored in screen coordinates, and depth position is stored in world coordinates. The 20 actions are chosen in the context of interactions with game consoles. As shown in Fig. 4, actions in this dataset reasonably capture a variety of motions related to arms, legs, torso, and their combinations.

Table 1: 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)

In order to facilitate a fair comparison with the state-of-the-arts, we follow the same experimental settings as [11, 25, 26] to split 20 actions into three subsets as listed in Table 1. In each subset, there are further 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 for training and the rest ones used for testing.

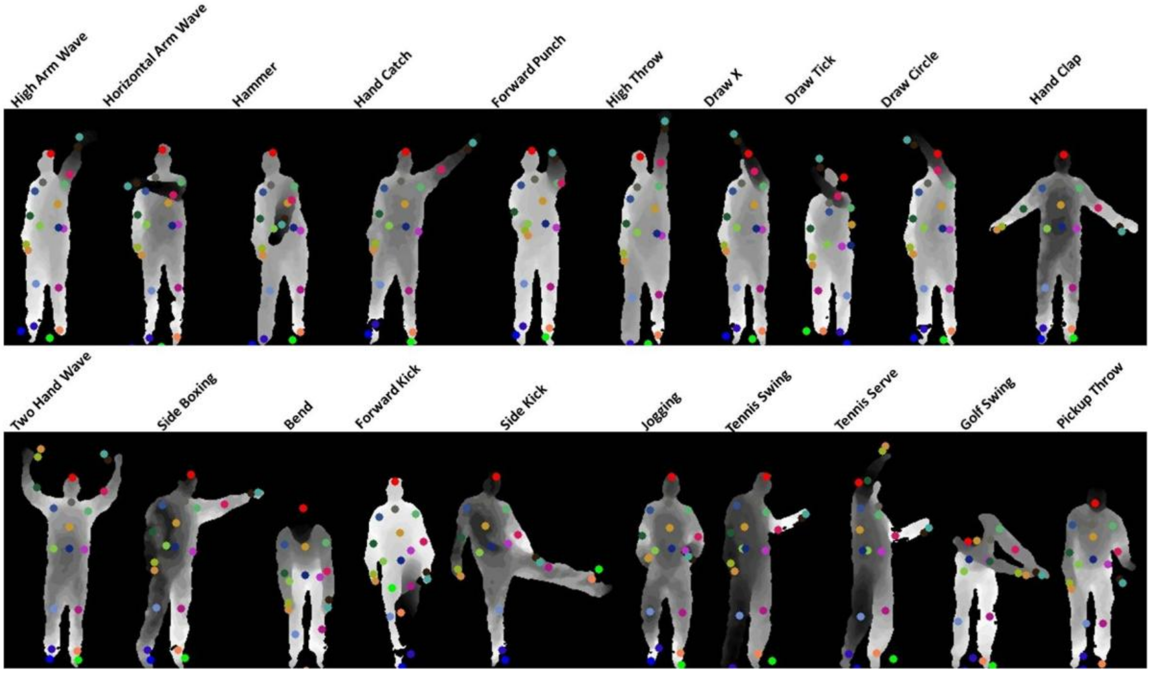


Figure 4: Examples of depth maps and skeleton joints associated with each frame of twenty actions in the MSR Action3D dataset.

6.1.1. Evaluations of EigenJoints and NBNN

We first evaluate energy distributions of joint differences to determine the dimensionality of EigenJoints. Fig. 5(a) shows ratios between the sum of first few leading eigenvalues and the sum of all eigenvalues of f_{norm} under different test sets. As demonstrated 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. 5(b) shows recognition accuracies of EigenJoints-based NBNN with different dimensions under various test sets. It’s interesting to observe that the overall recognition rates under a variety of test sets are stable across different dimensions. For each dimensionality, our method performs 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 motions, but AS3 groups complex but pretty distinct actions. For example, in AS1 *Hammer* tends to be confused with *Forward Punch*, and *Pickup Throw* consists of *Bend* and *High Throw*. In Cross Subject Test, different subjects also perform actions with considerable variations but the number of subjects is limited. 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 in more subjects.

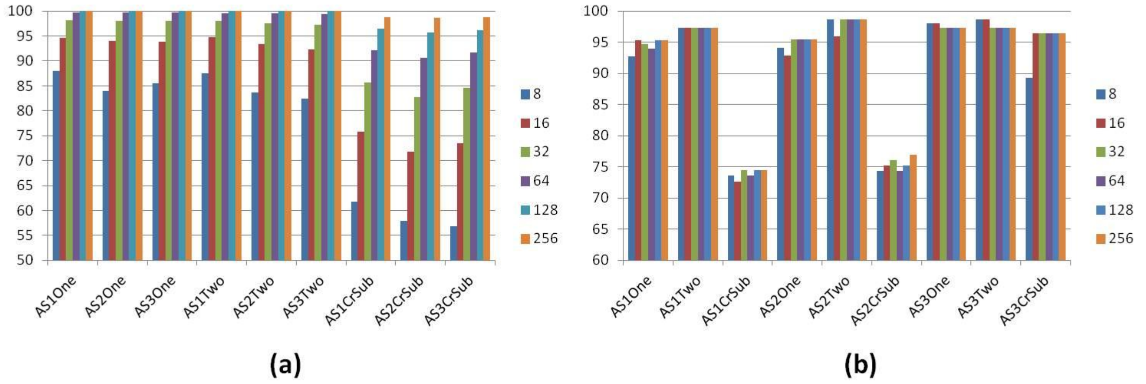


Figure 5: (a) Ratios (%) between the sum of the first few (8, 16, 32, 64, 128, and 256) leading eigenvalues and the sum of all eigenvalues of f_{norm} under different test sets. (b) Recognition accuracies (%) of NBNN-based EigenJoints with different dimensions under various test sets.

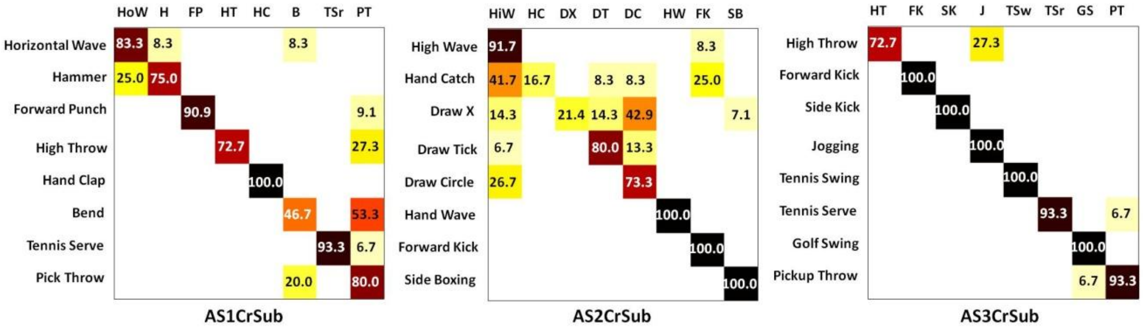


Figure 6: Confusion matrix of EigenJoints-based NBNN in different action sets under Cross Subject Test. Each row corresponds to ground truth label and each column denotes the recognition results.

Considering recognition accuracy and 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 Fig. 5), we only show the confusion matrix of our method under Cross Subject Test in Fig. 6. Because of the considerable variations in the same 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 mutually confused, as they contain highly similar motions. Although actions in AS3 are complex, they are with significant differences. So the recognition results are greatly improved in AS3CrSub.

6.1.2. Comparisons with the State-of-the-arts

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 = 500 and RBF kernels to perform classification. The optimal parameters of RBF kernels are obtained by 5-fold cross-validation. Fig. 7(a) compares the recognition accuracies based on NBNN and SVM. As shown in this figure, NBNN outperforms SVM in most testing sets. This observation also validates the superiority of the two schemes used in NBNN, i.e., non-quantization of EigenJoints and computation of *Video-to-Class* distance.

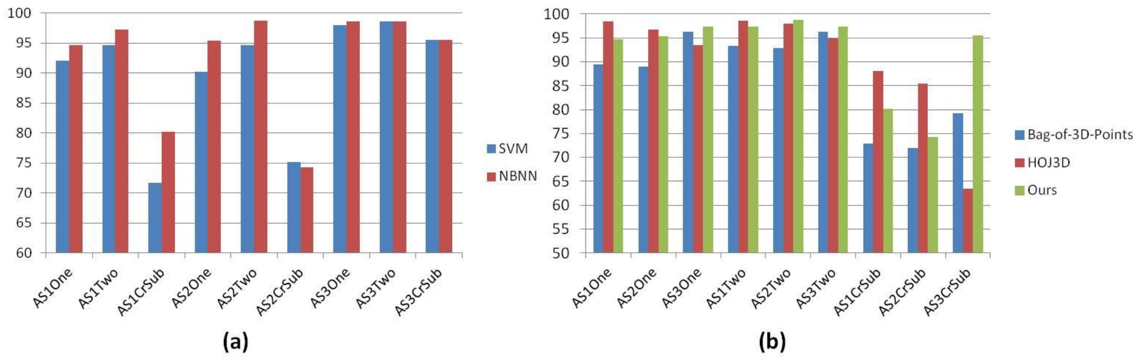


Figure 7: (a) Comparisons of recognition accuracy (%) between SVM and NBNN based on EigenJoints. (b) Recognition accuracies (%) of our method and the state-of-the-arts under a variety of testing sets.

We further compare our approach with the state-of-the-art methods including Bag-of-3D-Points [11] and HOJ3D [25] under different testing sets in Fig. 7(b). The overall accuracies are shown in Table 2. The results of Bag-of-3D-Points and HOJ3D are obtained from [11, 25]. As shown in Fig. 7(b), HOJ3D and our method significantly outperform Bag-of-3D-Points in most cases. The performances of our method are comparable to that of HOJ3D in non-cross-subject tests. However, under cross subject tests, HOJ3D and our method behave quite differently. Our method performs much better than HOJ3D in AS3CrSub, but is inferior to HOJ3D in AS1CrSub and AS2CrSub. This is probably because AS1 and AS2 group similar actions which are more sensitive to the larger intra-class variations generated in cross subject tests. So the leading factors computed by PCA might be biased by the large intra-class variations. But complex actions in AS3 present considerable inter-class variations which overweight intra-class variations. So the leading factors of PCA still correspond to variations of different action classes. As for overall accuracies in Table 2, our method and HOJ3D achieve comparable results in Test One and Test Two. But our method significantly outperforms HOJ3D under Cross Subject Test, which is more desirable in real applications. In addition to recognition accuracy, our method is more compact than Bag-of-3D-Points and HOJ3D. We further perform a more challenging experiment by combining subsets AS1-3 and obtain the accuracy of 74.5%. Wang *et al*. [24] achieved 88.2% accuracy by using multiple feature channels (skeleton joints and depth maps) and an actionlet mining algorithm. They observed the actionlet mining method was effective to handle noises and errors in skeleton joint positions when sever occlusion occurred. However, the multiple feature fusion and skeleton joint selection task are out of scope of this paper.

Table 2: Overall recognition accuracies of our method and the state-of-the-arts under three test sets.

Methods	Test One	Test Two	Cross Subject Test
Bag-of-3D-Points [11]	91.6	94.2	74.7
HOJ3D [25]	**96.2**	97.2	79.0
Ours	95.8	**97.8**	**83.3**

6.1.3. How Many Frames Are Sufficient

Both Bag-of-3D-Points [11] and HOJ3D [25] recognized actions using entire video sequences. We perform another experiment to investigate how many frames are sufficient to enable accurate action recognition in our framework. The recognition accuracies using different number of first few frames under a variety of test sets are illustrated in Fig. 8. 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, i.e., the first 30% to 40% frames are sufficient to achieve comparable recognition accuracies to the ones using entire video sequences. There are rapid diminishing gains as more frames are added in. These results are highly relevant for action recognition systems where decisions have to be made on line. An online system generally requires short latency that is mainly affected by two factors, i.e., 1) the time a system takes to observe sufficient frames for making a reasonable prediction and 2) the time a system takes to compute on the observations. Therefore cutting down the number of frames an online system reads in helps to reduce the costs in both of the two factors.

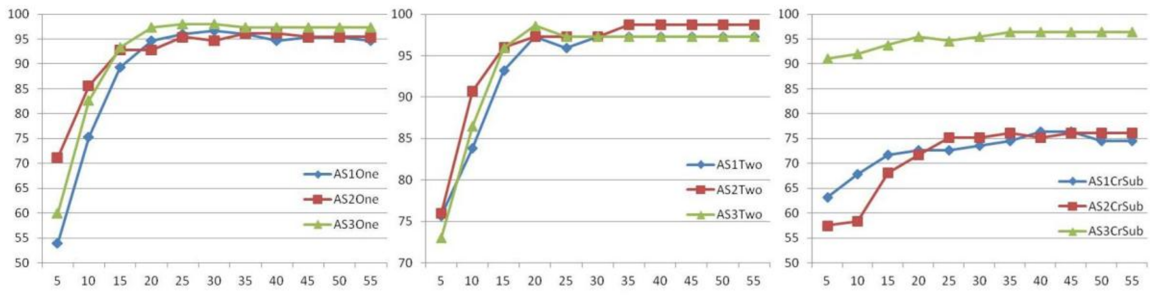


Figure 8: The recognition accuracies using different number of first few frames in Test One (left), Test Two (middle), and Cross Subject Test (right).

6.2. Experiments on the Cornell Human Activity Dataset

The Cornell Human Activity [23] is a public dataset that provides sequences of RGB images with aligned depth maps captured by a Microsoft Kinect camera. In each frame, 15 skeleton joints in world coordinates are available. Action videos are with the resolution of 640×480 and at the frame rate of 30 Hz. This dataset includes 12 activities and 1 random action performed by 4 subjects in 5 different environments (i.e., office, kitchen, bedroom, bathroom, and living room). The 12 actions are chosen in the context of human daily activities. As we can see in Fig. 9, activities in this dataset are captured in uncontrolled environments with cluttered households and involve extensive human-object interactions.

Table 3: Activities in five different environments.

Bathroom	Bedroom	Kitchen	Living Room	Office
Rinsing Mouth Brushing Teeth Wearing Contact Lens Random	Talking on Phone Drinking Water Opening Container Random	Chopping Stirring Drinking Water Opening Container Random	Talking on Phone Drinking Water Talking on Couch Relaxing on Couch Random	Talking on Phone Writing on Whiteboard Drinking Water Working on Computer Random

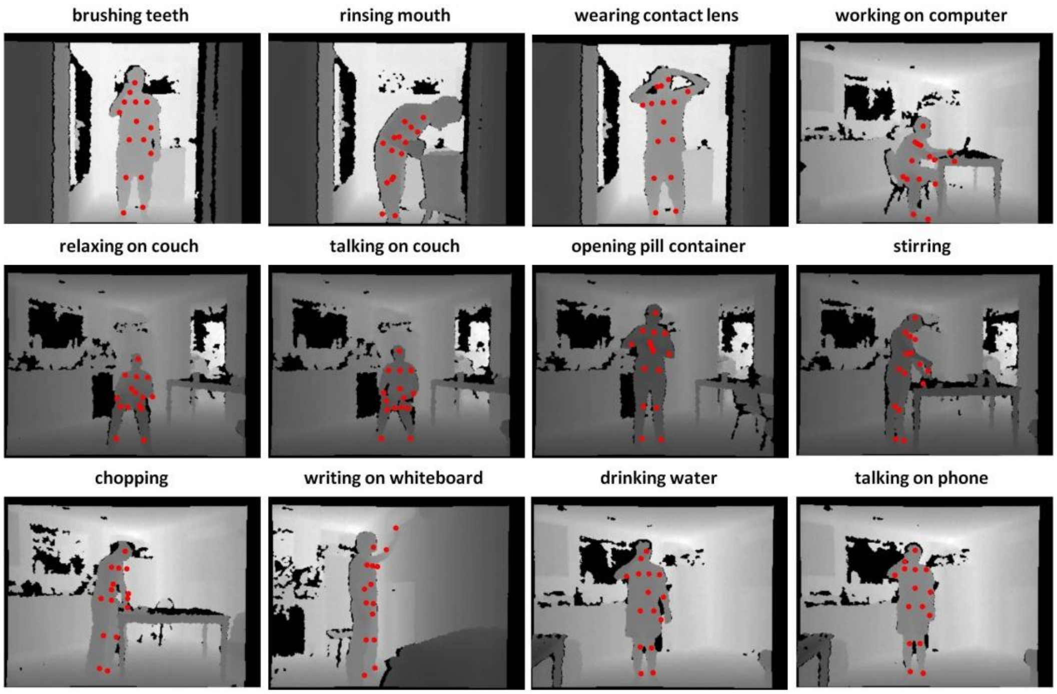


Figure 9: Examples of the depth maps and skeleton joints associated with each frame for twelve activities in the Cornell Human Activity dataset.

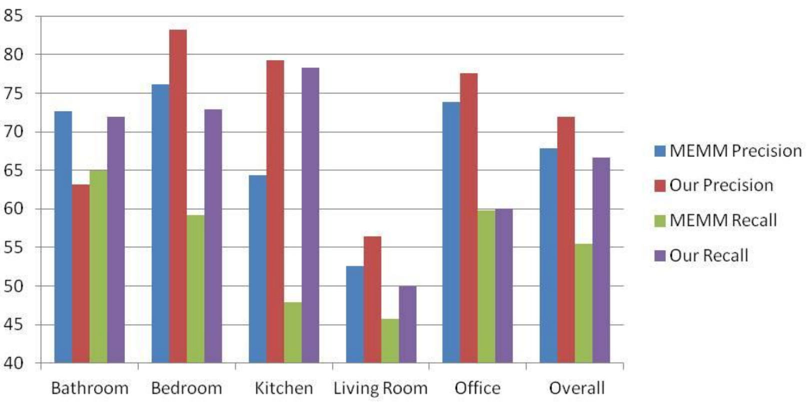


Figure 10: Precisions (%) and Recalls (%) of MEMM and our method under a variety of test sets.

Since neutral postures are removed in this dataset, we only employ f_{cc} and f_{cp} in Eq. (1-2) to compute EigenJoints. We follow the same experimental settings (subject independent test) as [23] to split the thirteen activities into five different environments under cross subject tests, as listed in Table 3. Experimental results are reported as average accuracies of leave-one-out tests, as shown in Fig. 10. The features used in hierarchical MEMM include visual (RGB) frames, depth (D) maps, and skeleton joints, which are much more complex than EigenJoints that only employs joints. However EigenJoints still significantly outperforms hierarchical MEMM, e.g., the overall precision and recall of our method are 71.9% and 66.6% which improves hierarchical MEMM's by 4.0% and 11.1%.

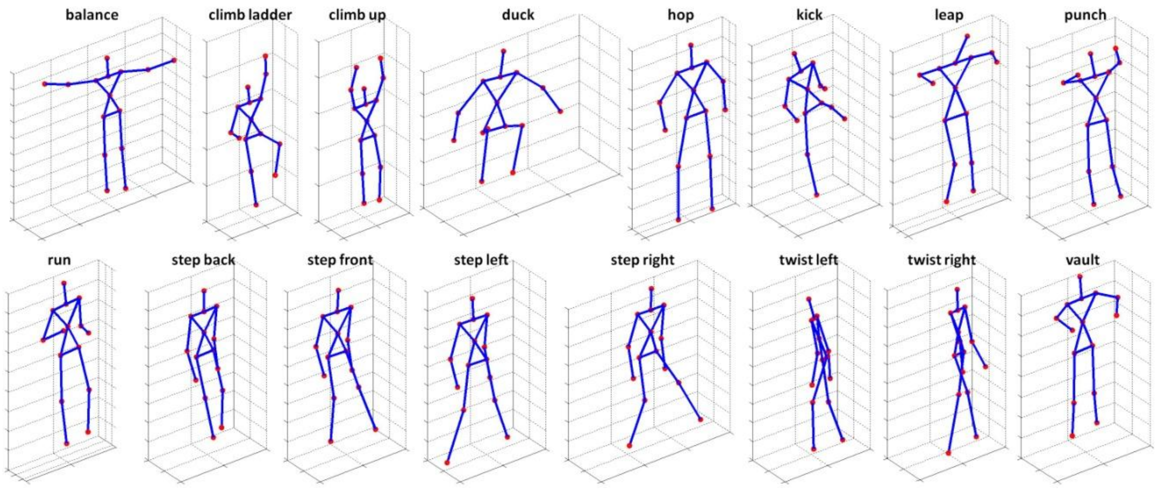


Figure 11: Sixteen actions and skeleton joints associated with each frame in the UCF Kinect dataset.

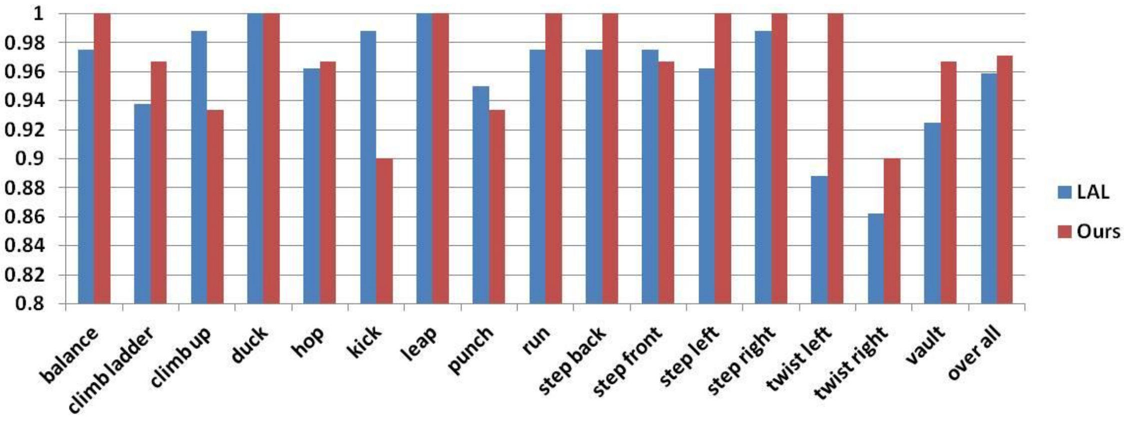


Figure 12: Comparisons of recognition accuracies (%) of LAL and our method.

6.3. Experiments on the UCF Kinect Dataset

We also evaluate our proposed method on the UCF Kinect dataset [14]. This dataset was collected by Microsoft Kinect and OpenNI platform. In each frame only 15 skeleton joints are available, RGB images and depth maps are not stored. It includes 16 actions performed by 16 subjects, as shown in Fig. 11. The comparisons of recognition accuracies of our method and the Latency Aware Learning (LAL) method [14] are shown in Fig. 12. Since depth maps are not available in this dataset, we do not perform frame selection but operate on entire video sequences. In order to reduce observational latency, the LAL method [14] aimed to search a single canonical posture for recognition. But to facilitate a fair comparison, we only compare to their results computed on full video sequences. It can be seen from Fig. 12 that our method achieves better or equal accuracies in 12 out of 16 action categories. The average accuracy of all the sixteen activities of our method is 97.1% which outperforms LAL by 1.2%.

7. Conclusion

In this paper, we have proposed an EigenJoints-based action recognition method using NBNN classifier. The compact and discriminative frame representation of EigenJoints is effective to capture the properties of static posture, motion between consecutive frames, and overall dynamics with respect to the neutral status. The proposed measurement of Accumulated Motion Energy (AME) quantizes the distinctiveness of each frame. By using AME to prune less discriminative frames, we can remove noisy frames and reduce computational cost. The comparisons between NBNN and SVM show that non-quantization of descriptors and computation of *Video-to-Class* distance are more effective for action recognition. In addition, we observe that the first 30%-40% frames are sufficient to enable action recognition with reasonably accurate results. This observation is highly relevant to the systems where action recognition has to be made online. The experimental results on three challenging datasets of the MSR Action3D, the Cornell Human Activity, and the UCF Kinect demonstrate that our approach significantly outperforms the state-of-the-art methods.

8. Acknowledgement

This work was supported in part by NSF grant IIS-0957016, PSC-CUNY #64720-00 42, and Microsoft Research. The authors thank the anonymous reviewers for their constructive comments and insightful suggestions that improved the quality of this paper.

9. References

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

[2] W. Bian, D. Tao, and Y. Rui. Cross-Domain Human Action Recognition. *IEEE Trans. on System, Man, and Cybernetics* (*Part B*), 2012.

[3] A. Bobick and J. Davis. The Recognition of Human Movement Using Temporal Templates. *IEEE Trans. on Pattern Analysis and Machine Intelligence,* 2001.

[4] O. Boiman, E. Shechtman, and M. Irani. In Defense of Nearest-Neighbor Based Image Classification. In Proc. *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[5] H. Cheng and J. Hwang. Integrated Video Object Tracking with Applications in Trajectory-based Event Detection. *Journal of Visual Communication and Image Representation*, 2011.

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

[7] R. Gishick, J. Shotton, P. Kohli, A. Criminisi, and A. Fitzgibbon. Efficient Regression of General Activity Human Poses from Depth Images, In Proc. *International Conference on Computer Vision*, 2011.

[8] H. Gunes and M. Piccardi. Automatic Temporal Segment Detection and Affect Recognition from Face and Body Display. *IEEE Trans. on Systems, Man, and Cybernetics* (*Part B*), 2009.

[9] A. Klaser, M. Marszalek, and C. Schmid. A Spatio-Temporal Descriptor based on 3D Gradients. In Proc. *British Machine Vision Conference*, 2008.

[10] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning Realistic Human Actions from Movies. In Proc. *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

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

[12] H. Liu, M. Sun, R. Wu, and S. Yu. Automatic Video Activity Detection Using Compressed Domain Motion Trajectories for H.264 Videos. *Journal of Visual Communication and Image Representation*, 2011.

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

[14] C. Ellis, S. Masood, M. Tappen, J. Laviola, and R. Sukthankar. Exploring the Trade-off Between Accuracy and Observational Latency in Action Recognition. *International Journal of Computer Vision*, 2012.

[15] S. McCann and D. Lowe. Local Naïve Bayes Nearest Neighbor for Image Classification. In Proc. *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[16] H. Pirsiavash and D. Ramanan. Detecting Activities of Daily Living in First-Person Camera Views. In Proc. *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[17] L. Han, X. Wu, W. Liang, G. Hou, and Y. Jia. Discriminative Human Action Recognition in the Learned Hierarchical Manifold Space. *Image and Vision Computing*, 2010.

[18] M. Roccetti, G. Marfia, and A. Semeraro. Playing into the Wild: A Gesture-based Interface for Gaming in Public Spaces. *Journal of Visual Communication and Image Representation*, 2012.

[19] K. Schindler and L. Gool. Action Snippets: How Many Frames Does Human Action Recognition Require? In Proc. *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[20] 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 Proc. *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[21] M. Sun, P. Kohli, and J. Shotton. Conditional Regression Forests for Human Pose Estimation. In Proc. *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[22] J. Sun, X. Wu, S. Yan, L. Cheong, T. Chua, and J. Li. Hierarchical Spatio-Temporal Context Modeling for Action Recognition. In Proc. *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[23] J. Sung, C. Ponce, B. Selman, and A. Saxena. Unstructured Human Activity Detection from RGBD Images. In Proc. *International Conference on Robotics and Automation*, 2012.

[24] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining Actionlet Ensemble for Action Recognition with Depth Cameras. In Proc. *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[25] L. Xia, C. Chen, and J. Aggarwal. View Invariant Human Action Recognition Using Histograms of 3D Joints. *IEEE CVPR Workshop on Human Activity Understanding from 3D Data*, 2012.

[26] X. Yang and Y. Tian. EigenJoints-based Action Recognition Using Naïve-Bayes-Nearest-Neighbor. *IEEE CVPR Workshop on Human Activity Understanding from 3D Data*, 2012.

[27] X. Yang, C. Yi, L. Cao, and Y. Tian. MediaCCNY at TRECVID 2012: Surveillance Event Detection. *NIST TRECVID Workshop*, 2012.

[28] X. Yang, C. Zhang, and Y. Tian. Recognizing Actions Using Depth Motion Maps based Histograms of Oriented Gradients. In Proc. *ACM Multimedia*, 2012.

[29] G. Yu, J. Yuan, and Z. Liu. Real-Time Human Action Search Using Random Forest based Hough Voting. In Proc. *ACM Multimedia*, 2011.

[30] Z. Zhang and D. Tao. Slow Feature Analysis for Human Action Recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2012.

[31] C. Zhang, X. Yang, and Y. Tian. Histogram of 3D Facets: A Characteristic Descriptor for Hand Gesture Recognition. *International Conference on Automatic Face and Gesture Recognition*, 2013.

[32] http://research.microsoft.com/en-us/um/people/zliu/Action RecoRsrc/default.htm

[33] http://pr.cs.cornell.edu/humanactivities/data.php

[34] http://www.cs.ucf.edu/~smasood/research.html