Open access • Journal Article • DOI:10.1007/S00371-018-1489-7

Improving bag-of-poses with semi-temporal pose descriptors for skeleton-based action recognition — Source link

Saeid Agahian, Farhood Negin, Cemal Köse

Institutions: Karadeniz Technical University, French Institute for Research in Computer Science and Automation

Published on: 01 Apr 2019 - The Visual Computer (Springer Berlin Heidelberg)

Topics: Cluster analysis

Improving bag-of-poses with semi-temporal pose descriptors for skeleton-based action recognition

Farhood Negin, Saeid Agahian, Cemal Köse

Noname manuscript No.
(will be inserted by the editor)

Improving Bag-of-poses with Semi-temporal Pose Descriptors for Skeleton-based Action Recognition

Saeid Agahian · **Farhood Negin** · **Cemal Köse**

Received: date / Accepted: date

Abstract Over the last few decades, human action recognition has become one of the most challenging tasks in the field of computer vision. Employing economical depth sensors such as Microsoft Kinect as well as recent successes of deep learning approaches in image understanding has led to effortless and accurate extraction of 3D skeleton information. In this study, we have introduced a novel *bag-of-poses* framework for action recognition by exploiting 3D skeleton data. Our assumption is that any action can be represented with a set of predefined spatiotemporal poses. The pose descriptor is composed of two parts, the first part is concatenation of the normalized coordinate of the skeleton joints. The second part consists of temporal displacement of the joints which is constructed with predefined temporal offset. In order to generate the key poses, we apply *K-means* clustering overall training pose descriptors of dataset. To classify an action pose, we train a *SVM* classifier with the generated key poses. Thereby, every action on dataset is encoded with key-poses histogram. We use *ELM* classifier to recognize the actions since it has been shown to be faster, accurate, and more reliable than other classifiers. The proposed framework is validated with four publicly available benchmark 3D action datasets. The results show that our frame-

S. Agahian

Department of Computer Engineering, Faculty of Engineering, Karadeniz Technical University, 61080 Trabzon, Turkey

Tel.: +90-462-3773167

E-mail: saeid@ktu.edu.tr

f. Negin

INRIA Sophia Antipolis, 2004 Route des Lucioles - BP93 06902 Cedex-France

E-mail: farhood.negin@inria.fr

C. Köse

Department of Computer Engineering, Faculty of Engineering, Karadeniz Technical University, 61080 Trabzon, Turkey

Tel.: +90-462-3773167

Fax: +90-462-3773412

E-mail: ckose@ktu.edu.tr

work achieves state-of-the-art results on three of the datasets compared to the other methods and produces competitive result on the fourth.

Keywords Skeleton-based · 3D action recognition · bag-of-words · key poses · Extreme learning machine · RGB-D

1 Introduction

Vision based action recognition has been extensively studied by many researchers due to its immense applicability essential for different areas including surveillance, smart home, human computer interaction, robot vision, augmented reality, and video summarization and indexing [2, 45, 61]. Despite extensive studies and remarkable progress in past few decades, action recognition still remains as a dynamic research field where a lot of problems need to be resolved. Among them, variability of view point, variation of speed, acceleration and body size of the subjects, intra class variation and inter-class resemblance of actions are the most important challenges. Moreover, in order to have generic solution for robust action recognition, temporal and spatial segmentation of action in videos, semantic parsing of the action and sub-actions as well as collecting enough training data are another challenges that need to be addressed [48]. The conventional approach for action recognition task is to first extract hand-crafted features of different modalities (such as RGB, skeleton joint position, or depth map [46]) and then to classify the videos based on the calculated feature vectors [45]. An action can be described in three levels: low-level, mid-level, and high-level [16, 56]. In most of the early works, posture has been used as a high-level descriptor to describe human pose and their concatenation along joint trajectories for action recognition. However, difficulty of body part detection and reliable pose recovery and its high computational cost forced researchers to choose an alternative track [16]. Preliminary studies have led researchers to introduce low-level features which have been extracted either sparsely or densely from RGB videos. Human body silhouettes is one of the early examples of these type of features which has been widely used for human action recognition in environments where the background subtraction is applicable [5]. For the First time Laptev and Lindeberg extended the Harris edge detector into 3D space and produced sparse feature called *Spatio-temporal interest points (STIPs)* for action recognition [30]. The introduction of this feature led to vast success of adopting *bag-of-words* method which was already used in text processing, to recognize action from video [31]. Optical flow is another low-level feature that has been used by Efros et al [15] to describe and recognize human action. Meanwhile, the success of some of image based features in image classification encouraged the researcher to exploit them for video classification tasks. Among them *HOG3D* [50] and *SIFT3D* [27] have been used

for action recognition. However, extraction of low-level features is not limited to RGB data. For example, in [70], the provided depth images are considered as intensity images and utilized for low-level appearance-based feature extraction. Using bag-of-word methods along with low-level features comes with some limitations. The main drawback is its restriction to represent the spatial and temporal relations between the features [28]. In order to overcome this limitation, some researcher proposed mid-level features to model temporal and spatial dependency between low-level features. For example [64, 65] proposed semantic structure as motion trajectories instead of key-point to describe local motion of action.One major disadvantage of methods that use low-level and mid-level features is their inability to represent complex activities due to their limitation in presenting semantic information [49]. One possible solution came with introduction of high-level semantic features [16] where description of an action carried out using a sequence of semantic lexicon that encapsulates spatio-temporal body pose information. Accordingly, Microsoft Kinect sensor made cost-efficient high-level markerless real-time pose extraction from RGB-D images available [21, 53] , which was a challenging problem for a long time. Lately, with resurgence of Deep Learning methods, precise and reliable body pose recovery from RGB images is provided with low cost and is not limited to depths sensors anymore [6,12]. Considerable progress has been done in accurate markerless pose detection awarding advantages such as its resistance to variation in view point, scale and appearance of subject for describing an action compared to low and mid level features. These privileges attracted researcher to focus more on these kind of input and use it extensively for feature extraction tasks [16]. The main challenge to use these data for action recognition is their heterogeneous numeric representation of semantically similar actions.Wang and et al [63] divided the proposed pose-based approach into three categories in terms of modeling temporality in actions. Methods in the first category ignore temporal dependency information and treat each pose in the sequence individually [4,19,59] e.g. [19,59]adapt *bag-of-poses* for describing the actions and uses majority voting [4] to carry out classification. Ben-Arie, et al [4] assume that the entire action could be recognized by only having specific poses extracted from the complete video frames. The second category consists of methods that exploit all of the available poses in the sequence to model the action and thereby, to classify it. Methods based on *Hidden Markov Models* [18, 69] or *dynamic time warping* [51] are the most prominent approaches in this category. In the third category there are methods that model the temporal structure of action by using pose information partially. For example, [37, 67] have used temporal pyramid matching and [62] has modeled the change of neighboring poses to maintain temporal information.In recent years, *recurrent neural networks (RNN)* and *Long Short-Term Memory (LSTM) networks* have

achieved remarkable success in text and sound recognition for modeling temporal dependencies in sequences [32]. Nevertheless, there are a few proposed methods that use variation of *RNN* [58] and *LSTM networks* [80] with skeleton data and achieved acceptable results. One of the main challenges of using Neural Network and Deep Learning for 3D action recognition is a lack of training data. Moreover, computational complexity of these networks makes it unsuitable for use in real time and online tasks [20,38].*Generative* methods (state-based) such as *HMMs* produced acceptable results for modeling action with pre-defined poses [69]. But the main disadvantage of these method is their sensitivity to training data where only abundancy of data in training phase may lead to performance enhancement [48]. Moreover, training of *HMMs* in terms of computational and memory cost is expensive and requires manual parameter tuning. Therefore, using *HMMs* with noisy skeleton data generally does not end up producing excellent results since it is difficult to determine a correct state where there are some variation in candidate actions.On the other hand, instead of generative models, discriminative methods such as kernel machines or metric learning that have been developed for classification of vector data are more suitable for working with high dimensional space [13]. These methods generally have achieved better results compared to *HMMs* [29] and have been used for recognition of single action in pre-segmented video clips. Conversely, generative methods have been used for parsing and segmentation of continuous videos. As outlined before, one of the main limitations in bag-of-pose methods is to ignore the concept of time and modeling the relationship between the poses. This means that the order of poses which is an important aspect of modeling an action, is neglected in learning phase. There have been a lot of effort in the literature to preserve temporal information. Among the others *temporal pyramid* [75] and producing histograms for distinctive segments of actions gained more popularity. Moreover, some method tries to add temporal features such as speed to describe each pose while keeping temporal information [16, 62]. Considering the abovementioned discussions and with presumption of reliability of the 3D poses and knowing that they do not require sophisticated feature modeling and learning [16], in this work, we propose a pose-based action recognition framework.

The main idea is to describe an action with a set of predefined poses and to encode it by histogram of those poses. Fig.1 illustrates the overall data flow of the proposed method. We describe poses of a sequence by defining a simple and efficient temporal and spatial feature. Using this feature enables us to distinguish between two actions with same skeleton configurations but different temporal order of their key poses. According to the conducted experiments, the proposed descriptor produces more discriminative key poses from training poses for action representation. Embedded temporal information in the key poses helps us to overcome the limita-

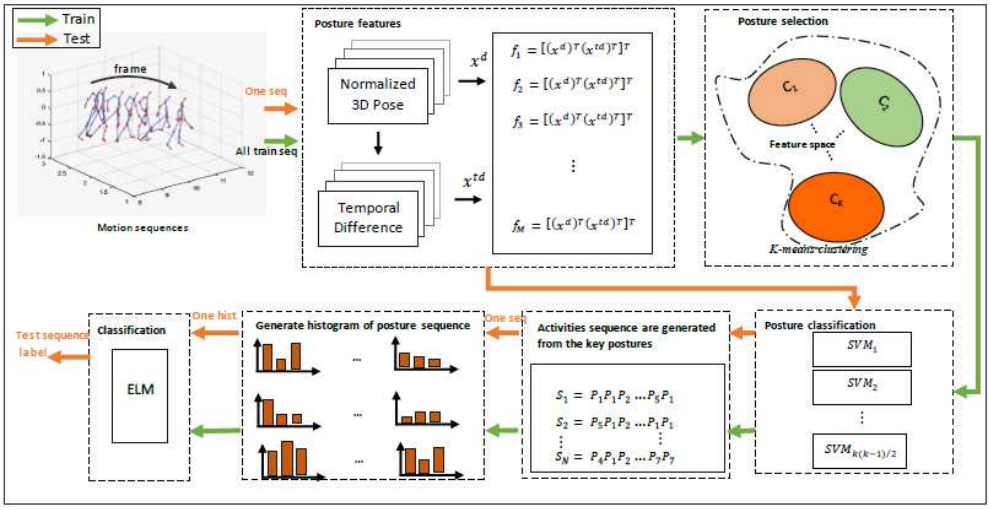

Fig. 1: Workflow of our method

tions available in bag-of-word methods by encoding action with histogram of key poses. The length of feature vector that describes the actions is fixed and independent from total number of frames. Finally, we used discriminative Extreme Learning Machine [23] for classifying the actions which is expressed by the key pose histograms. We have tried the proposed methodology on four publicly available benchmark datasets that include 3D skeleton data. Based on the obtained results, we show that our method is capable to produce state-of-the-art results compared to the other methods on three of the datasets only by using skeleton data and competitive result on the fourth dataset. Simplicity, Interpretability and high processing performance in recognition are the major advantages of the proposed methodology. This paper is organized as follows: in Sect.2 there is a brief explanation of available approaches in the literature. The detail of our approach is described in Sect.3. The experimental evaluation and results on four public datasets are presented in Sect.4 and finally, we conclude and summarize the paper in Sect.5.

2 Related work

In this section, we briefly explain pose-based methods that solely employ 3D skeleton data for action recognition. It should be noticed that the 3D action recognition is not limited to articulated pose-based methods that use 3D skeleton joints, therefore, for more details on this topic, we invite the readers to refer to the surveys in [2, 45, 75]. The 3D skeleton data represents relations between the joints and overall configuration of human pose. This information can be extracted from different modalities such as motion capture systems (*MoCap*), stereo and range sensors [1, 20], etc. As a pioneering study on human action recognition, Johansson [25] showed that availability of the joint position sequence is sufficient to recognize human actions. Yao et al.

[72] showed that in indoor action recognition scenarios, using pose-based features results in a better recognition performance compared to those in appearance-based features. Employing skeleton data for action recognition has many advantages over other modalities including insensitivity to variability of scale and illumination, independency of viewpoint position of the subjects and speed of their action. It also increases data processing speed and efficiency which makes it a suitable candidate for online and real-time applications [20]. Additionally, body part information provided by skeleton data can underline the parts that have more contribution in human actions and offer more interpretation capabilities [55, 60]. However, there are some disadvantages and restrictions in using skeleton data such as failure to transfer both contextual information and objects around the joints which are necessary for recognition of human-object interaction and Human-Human interactions [1, 3]. In general, all pose-based action recognition approaches consist of two major steps: first, human poses in each frame are described by the features extracted from raw 3D skeleton data and afterwards, the final feature vector is calculated for whole action sequence to be used in classification or reasoning. Han and et al. in their recent work, name the first step as Information modality and the second step as Representation encoding and group different methods available in the literature into each step [20]. According to this taxonomy, various CD pose-based features which has been used for describing actions are categorized into four groups:

Displacement-based representations: Applying this method needs low computational cost and has a simple structure. These features are usually extracted in two ways; in the first type, displacement between each pair of skeleton joint positions in the same frame is calculated, and then, in the second, displacements among the corresponding joints in two different frames are computed. Therefore, these features can describe displacement as spatial or temporal signature of an action. Spatial form of these features was used in studies done by [67] and [66] and displacement features were normalized to make them invariant to subject scale. In another type of spatial displacement feature, the features are generated using displacement information between each joint and a selected reference joint. Hip center joint has the least movement compared to other joints during an action. It is therefore considered as a reference joint in different studies [3]. Often times, using only spatial displacement features are not sufficient to thoroughly describe dynamics of an action, and consequently, temporal features are proposed and used as complementary features. The most common temporal features are generated using displacement position information of joints in two different frames. In order to do this, the reference frame is selected as either previous frame or as natural pose which is average of the initial frames in overall instances. Apart from this, the reference frame can vary in course of an action and can be identified with a time off-

set [11].

Orientation-based representations: Joint orientation is one of the common features that have been widely used for pose description owing to its inherent invariance towards human position, body size, and orientation of the camera. There are two types of these features: The spatial orientation feature that is computed from orientation of the pairwise displacement vectors of joints at the same frame [69]while the temporal orientation feature is computed by considering the orientation of displacement vectors of the same joint at two different frames.

Representations based on raw joint positions: Along with the displacement-based and orientation-based features, raw joint positions has been extensively used in many studies. A group of methods concatenate all joint positions of the skeleton at a frame into a vertical vector and put them together to construct a matrix to encode the action. Obviously, in the constructed feature matrix of an action, joints have different representative qualities and importance. To select the most informative subset of the joints for an action, different methods have been examined. Among them, Chaaraoui and A.A. [7] exploited evolutionary algorithm to find the optimal subset based on topological information of the skeleton and achieved state-of-the-art results. Raw joint positions require normalization processing to be independent of scale and rotation. Another group of these methods constructs joint trajectories instead of raw joint position information and then define the action via extracted features [3]. Veeriah et al. [58] and Zhu et al. [80] employed raw joint positions as input of the deep networks and let the network to discover the best representation using this information. This is similar to conventional deep learning methods that automatically learn representations from pixels of input images.

Multi-modal representations: To achieve an accurate and more powerful action representations, combination of different features can be utilized [16, 59]. In particular, combination of time and space features has gained more attention and has obtained better results [34, 74]. In order to find the optimized combination of available features, Negin et al. [41] introduced a discriminative feature selection framework based on Random Decision forests to identify the most effective subset of available feature types in space and time. Considering the above-mentioned discussion, the goal of feature representation and encoding is to integrate all of the extracted features and generate the final feature vector which will be used in classification or reasoning phase.

The encoding methods are categorized into three main groups [20]:

Concatenation-based approach: It is the most straightforward encoding method which is carried out by simply concatenation of the extracted features into a one dimensional final feature vector [17, 66]. The generated feature vector is too long and is therefore practically difficult for classifier to handle the high dimensional space. To reduce dimensions of

the feature vectors, dimensionality reduction methods such as *PCA* or *LDA* can be utilized which leads to an increase in computational cost [69].

Statistics-based encoding: This is one of the common and efficient methods for integrating the features which is mostly performed by applying statistical analytics on constituent feature vectors without using any feature quantization operation. For example, Hussein and M.E. [24] proposed *Cov3DJ* descriptor constructed from covariance matrix of the sequence vectors. The sequence vectors are composed of concatenated joint positions at each frame. One of the advantages of Statistics-based methods is that the length of the final feature vector is independent of the number of action frames. One dimensional histogram-based encoding has been used more than other encoding methods in this category [69]. However, a lack of order in feature elements and absence of temporal relation can be considered as the most important drawbacks of these methods.

Bag-of-words encoding: These methods employ coding operator and dictionary learning for mapping a high dimensional feature vector into a single code-word in a dictionary. Using coding operator for quantization of the feature vectors is the main difference between these methods and the Statistical methods. Han et al. [20]extracted different features from skeleton data and applied each of these three coding methods to the obtained feature vectors. Their results indicated that the *Bag-of-words* encoding methods gave a better performance compared to the other ones on four benchmark datasets that they evaluated.

In terms of dictionary learning, the encoding methods are generally divided into two main categories: clustering and sparse coding based methods [20]. *K-means* is a widely-used clustering method for generating a dictionary. Similar to other *bag-of-words* methods, losing temporal information among the features is a major shortage of this method. There are some studies [26, 62, 74] in the literature that have been conducted to overcome this deficit and improve reliability of the encoding methods. In order to extract spatial/temporal structure of the poses in each action class, Wang et al [62] used Contrast Mining techniques for grouping skeleton joints in training data followed by *K-means* clustering over each group. To learn the dictionary pertinent to each group, they used cluster centers as the code words which encodes the spatial information of the action. For encoding the temporal structure of each action class, Contrast Mining technique was used. This technique extracts sub-sequences that occur frequently among sequences of each group. The spatial/temporal histogram of skeleton joints was used for action representation and one-vs-one learning techniques was applied on pairwise classes for classification. This method benefits from a pose recovery technique that helps to improve pose detection from images, however, applying data mining on both of the encoding steps leads to a high computational cost. Zanfir et al [74] introduced a new type of fea-

ture called moving pose feature which includes both temporal and spatial information. To describe pose in each frame, it uses raw 3D joint positions along with the first and second derivatives of the joint trajectories. Then, distinctive poses are selected by data mining methods. A modified version of *KNN* classifier was used to classify the test instances. Temporal pyramid is one of the alternatives for representing temporal information in *Bag-of-words* methods. To describe the poses in each frame, Eweiwi et al. [16] concatenate relative location of the joints, velocity, and their correlation and use it as a feature vector. Instead of one histogram for all action frames, they represented the actions by a two-layer pyramid histogram. One histogram from all action frames is computed in the first layer and later, in the second layer, all frames are divided into three equal sections and then, a histogram is computed from each one of the sections. The final action descriptor is constructed by concatenation of the four histograms and classification is performed by applying *Kernel-PLS (KPLS)* on the feature vectors. Kapsouras and Nikolaidis [26] used joint orientation and their differences in three various periods of time as features and accordingly, they applied a modified *Bag-of-words* strategy to represent the actions. For pattern recognition, first, the *K-means* applied on the sets of features individually and one histogram is generated for each set. Then, the whole action is represented by concatenation of these four histograms.

In training step of the successful studies, selecting representative features is performed via expensive computational methods such as data mining or other feature selection mechanisms [16]. Providing spatial/temporal information using these mechanisms for the *Bag-of-words* methods is accompanied by a higher level of complexity. The most similar *bag-of-words* method to our study is [36].

In this study, for calculating temporal displacement pose descriptors at frame t with a randomly preselected differential time offset Δt, for each element i, they obtain $\triangle\theta^i = (x_t^i - x_{t-\triangle t}^i, y_t^i - y_{t-\triangle t}^i, z_t^i - z_{t-\triangle t}^i)$. Accordingly, feature vector is constructed by concatenating the calculated $\triangle\theta^i$ for each element $(i \in 1,2,\cdots,m)$. *K-means* is applied on the extracted pose descriptors on training data and encoding is performed by finding the closest cluster center to the obtained pose word. Before feeding the descriptors into a Nave Bayes voting based classifier, each part of the motion is separately encoded followed by generating a histogram specific to each part. The main difference between this method and ours appears in pose encoding phase which was conducted in low-level and high-level pose encodings, respectively. Each word in our method describes a real pose while in [36], a word is a directed vector describing each local part. Our descriptor is effective as it produces low dimensional feature vectors which is independent of the number of the skeleton elements and only depends on the number of the key poses. [36] ignores spatial information while our method uses spatial information along with temporal pose data. Nowadays, due

to the extensive progress in image processing and classification by using deep learning methods such as *Convolutional Neural Networks (CNN)*, researchers have been encouraged to employ it for skeleton-based action recognition. However, there are still some challenges that need to be resolved. These methods are designed to accept images as input and cannot capture the dynamic information in skeleton sequences. Therefore, an encoding method including spatio-temporal information of a sequence in two dimensional image space is required. Some of the works in the literature suggest to convert skeleton pose sequences into an image containing dynamic information and then, ask the network to classify the synthesized images. For example in [22], Hou et al. propose a new encoding method called *Skeleton Optical Spectra (SOS)* that transforms skeleton sequences into texture images. The generated textures are used as input to a *CNN network* to extract separable features where the classification is performed using the average output of the *CNN network*. Our proposed approach in this study is a pose-based method which utilizes *Bag-of-words (bag-of-poses)* method for encoding. Using simple features extracted from raw joint positions of the skeleton data distinguishes our method from other existing ones. These features are extracted directly from the raw joint positions without transforming them to another space such as Lie Groups [59,60]. The temporal information are embedded into the *Bag-of-words* dictionary without using complex data mining methods [63]. This is performed by generating spatial/temporal poses as words of the dictionary. Therefore, the generated histograms inherently contain the temporal information and using multiple histograms is not required for handling time information.

3 Proposed method

As input, the proposed framework accepts a sequence of high dimensional vector of skeleton joints for each action: $\left\{P^1, P^2 \cdots, P^m\right\}$ where m is frame number and each P in the sequence equals to a set of three-dimensional coordinates of skeleton joints at frame t:

$$P^t = [x_t^1, y_t^1, z_t^1, x_t^2, y_t^2, z_t^2, \cdots, x_t^n, y_t^n, z_t^n]$$

(where n is the number of skeleton joints). The coordinate system of the framework (x, y, z) is defined based on the location of the camera and as it is shown in is defined based on the location of the camera and as it is shown in Fig.2. its center matches with the center of the camera.

Inspired by conventional *Bag-of-Words* methods, the proposed method describes an action as a sequence of *pose-words (key pose)*. Encouraged readers can refer to [44] that has compiled a comprehensive survey summarizing *Bag-of-Words* methods which have applied on action recognition problem. The overall flows of the frameworks are shown in Fig.1. A preprocessing step precedes feature extraction process to make the input skeleton information invariable to subject position, scale and camera view.

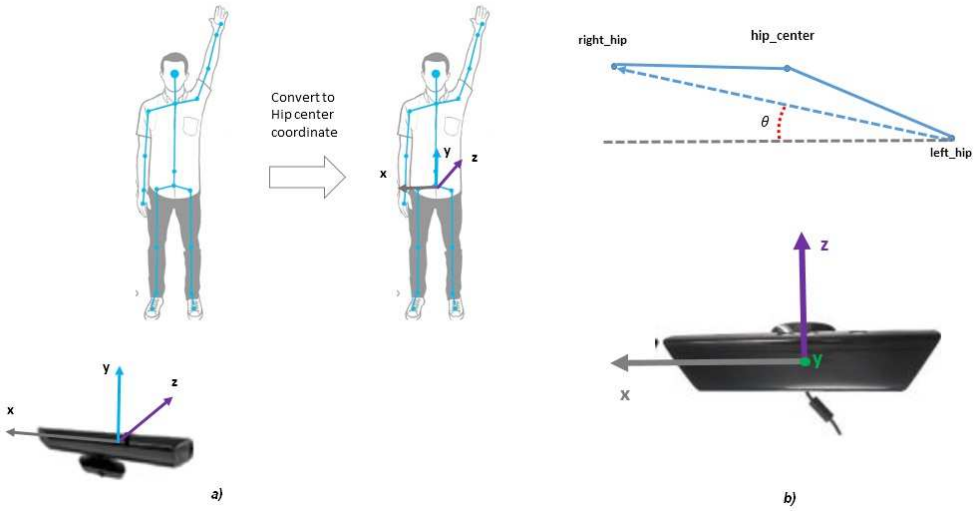

Fig. 2: a) Setup of Kinect Coordinate b) Rotation skeleton towards Kinect

3.1 Preprocessing and Feature Extractione

The preprocessing step makes the input skeleton data:

Transform invariant:in each frame, we transform the origin of the coordinate system from real-world coordinates to hip center of the person. This transformation makes the position of the skeleton joints invariable to the location of the subject. This transformation is demonstrated in below, where i is joint index.

$$(x_i^{'}, y_i^{'}, z_i^{'}) = (x_i - x_{hipcenter}, y_i - y_{hipcenter}, z_i - z_{hipcenter}) \quad (1)$$

Scale invariant:in general, people performing an action have a diverse ranges of body sizes. In order to have robust action models, the generated action features of different subjects should preserve consistency among the representations. Different methods have been proposed in the literature to maintain scale invariability, among them we use a method similar to [60]. First, we choose a random pose as reference and then, we rescale all the remaining poses limb sizes to the size of corresponding body parts in the reference pose preserving the original angles between the pose parts.

Rotation invariant: To make skeleton joints invariant to camera view, a specific rotation is performed with respect to specified view point of the camera. As shown in Fig.2b, this transformation makes sure that the projection of the vector passing from left hip to right hip on ground plane to stay parallel with x axis in real world coordinateswhere the rotation angle is computed by:

$$\theta = tan^{-1}\left(\frac{z_{rhip} - z_{lhip}}{x_{rhip} - x_{lhip}}\right) \quad (2)$$

After obtaining the deviation angle, we use Eq.3 for each skeleton joint in corresponding frame to rotate around the y axis in a counterclockwise fashion.

$$\begin{bmatrix} x_i' \\ y_i' \\ z_i' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ z_i \\ 1 \end{bmatrix} \quad (3)$$

Given a normalized pose, the next step is generating a pose descriptor. I. Lillo [34] classifies features of pose descriptors in two categories:

i. Geometric descriptor: Geometric descriptors represent the spatial configuration of the skeleton joints in each frame. These type of descriptors could use calculated angle between the skeletal vectors or the computed distance between joints using different metrics.

ii. Motion descriptor: Although the geometric descriptors are capable of defining spatial configurations of skeleton joint, they are unable to encode dynamic information of the poses. In order to codify motion dynamics in representation of poses motion descriptors utilize information such as velocity, speed, derivation and optic flow in their calculations. Motion descriptors also avoid the ambiguity between two poses where they have similar spatial configuration but embody different action characteristics(Fig.3).

While the proposed descriptor intrinsically contains geometric information, it tries also to keep track of dynamic of actions by taking into account temporal dependency between consecutive frames. The final pose representation could be composed of different combinations of the descriptor types. One popular combination strategy is to concatenate all of the extracted features, but as dimension of the descriptor increases the cost of classification is also proportionally scales up. In order to keep the dimensionality manageable most frequently dimensionality reduction procedure such as *PCA* or *LDA* are used. Although dimensionality reduction brings efficiency to processing of the descriptors, it is computationally expensive and sometimes it doesnt culminate the accuracy [18]. An alternative strategy that is called feature engineering, rather than blind concatenation of features tries to single out the most representative ones in the feature set. Feature engineering is usually done either manually (handcrafted)or automatically (learning-based) e.g. supervised sparse dictionary learning, neural network, Genetic programming (GP), CNN or Random decision forests [78]. Since feature selection mechanisms are computationally expensive, they cannot be a good choice for a real-time application [20]. Our features are similar to the one in [11] and gives an efficient pose description. As it is illustrated in Fig.1, to describe spatial configuration of skeleton in each frame we define the feature vector for kth frame as

$xd_k = [x_k^1, y_k^1, z_k^1, x_k^2, y_k^2, z_k^2, \cdots, x_k^n, y_k^n, z_k^n]$, that concatenates normalized coordinate of skeleton joints (n is number of joints in skeleton). As mentioned before, in order to model the temporal dependency between the poses in different frames and make the descriptors to embody information of similar action configurations but with composite temporal de-

Fig. 3: a) Sit Down and b) Stand up

pendencies, we define another vector xtd_k that models the temporal dependency by taking into account a randomly selected frame offset(k') :

$$xtd_k = \begin{cases} xd_k & 1 <= k < k' \\ \frac{xd_k - xd_{k-k'+1}}{\|xd_k - xd_{k-k'+1}\|} & k' <= k < K \end{cases} \quad (4)$$

If the current pose occurs before the key pose the calculated vector would contain regular joint features, otherwise, it also calculates the distance between the current pose and all of the dependent poses in the temporal offset from the key pose in the action sequence. K is the number of frames in sequence of an action and k' is a temporal fixed offset of the current frame. The final feature vector is composed of $xd_k = [(xd_k)^T, (xtd_k)^T]^T$,which is concatenation of spatial-temporal features, and its dimension $D = 3 * n * 2$ is linearly dependent on the number of skeleton joints.

3.2 Key poses selection

Similar to the *bag-of-words* methods, our framework represents a sequence of an activity with a set of initially learned key poses (words in the dictionary). Therefore, the dictionary of the key poses needs to be learned and subsequently high dimensional pose features in the sequence get encoded into single word. Conventionally there are two ways to learn the dictionaries:

i. The first way is to divide the features space into sub regions and then express each region with its representative (the code-word). *K-means* algorithm is widely used for this purpose [18, 26, 62].

ii. The second way is to determine the distribution of the features using a generative model. *Gaussian Mixture Model (GMM)* is the most popular method which has been used in this regard. The *K-means* algorithm generates the words from feature vectors based on hard assignments (i.e. uses Euclidean distance to find the closest center), while, *GMM* performs soft assignment instead (i.e. for code-words assignment, rather than mean value it uses probability distribution of the features) [44]. The accuracy of classification is directly related to the quality of the trained dictionary and encoding of the features. In case of *K-means* algorithm, as dimensionality of the feature vectors increases, Euclidean

distance performs poorly and starts to generate unreliable encodings. So, to improve dictionary learning and encoding, we perform it in two steps(Fig.1)similar to [18]. To generate pose words for the dictionary (Key poses) the *K-means* algorithm is applied on feature vectors of all the training frames(Eq.5):

$$P = \{\bigcup_{i}\bigcup_{k} x_k(i) | i \in 1,2,...,TrainingTrials \quad and \quad k \in 1,2,...,lengthFrames(triali)\} \tag{5}$$

Consequently, the feature space gets divided into a k clusters and their corresponding cluster centers. The obtained cluster centers are considered as key poses and passed to the next step of the framework.

3.3 Pose classification and encoding

To resolve the problem of Euclidian distance in the encoding phase, we train a set of *SVM* classifiers using the key pose of the dictionary and carry out the assignments using classification. For implementing, we use *LIBSVM* [8] library in which one-against-one method is used for classification of the key poses. We train $S = \frac{K*(K-1)}{2}$ binary *SVMs* for classification of K poses. For assignment of the features vectors to the key pose we use learned binary *SVMs* with *"max wins"* voting strategy. Using hyperplanes for classification of pose words yields better assignment results than *K-means* in the assignment [18].

3.4 Action representation using key poses histogram

In this step, we use the trained *SVM* classifier to convert each actions feature vector into a sequence of key poses. The sequence of produced poses has a variable length due to variety of frame number in the videos. For classification of variable length sequences most often methods such as Hidden Markov Model, Bayesian Network and Dynamic Time Warping are used [45]. Therefore, for classification of the activities we can use discriminative classifiers such as *SVM*, *KNN* and *ANN*. Normalizing the length of feature vectors to the fixed length is usually done in two ways: sampling the video frame to the desired size and then extracting of the features vector. The other quantization values of feature vectors to specified number and then use the histogram of quantized values to describe the entire action [48]. We described each activity with a fixed length feature vector, we calculate the histogram of the sequence containing the constituent key poses. Prior to these calculations, the length of histograms is determined with number of extracted key poses.

3.5 Action classification

There are several popular classifiers such as *KNN, SVM, ANN* and random forest for classification of fixed length feature vectors. In this work we use *Extreme Learning Machine (ELM)* classifier [23] in order to classify action. *ELM* is a single-layer feedforward neural network classifier which is

successfully applied in various applications and has shown high learning speed and viable accuracy. For the first time Minhas, R., et al [39] used this classifier on motion based features to detect human actions and it showed promising results. Moreover, this method is not only limited to low class number and small-scale classification, but also for classification large-scale realistic tasks. Varol, G. and A.A. Salah [57] used *ELM* for action recognition of realistic video clips and achieved acceptable results taking into account heavy computational cost of deep neural network methods. In recent years, this method also has been used to detect human action with skeleton data [11, 73].

4 Experimental evaluation and results

We evaluated our method on four challenging benchmark datasets. We assumes that there is only one person performing the assigned actions. This explains that why we observe a drop in performance when interactive actions are evaluated.

UTKinect-Action dataset:

UTKinect-Action [69] dataset is collected by Xia, Chen, and Aggarwal at the University of Texas at Austin in 2012. The data captured by Kinect v1 in 30 fps and includes 10 actions. Each action performed by 10 subjects (9 men and 1 woman) for 2 times. In total 200 sequences exist in the dataset. The dataset includes RGB, depth, and skeleton where the sequences are manually clipped. Similarly, skeleton data in each frame is represented by Euclidean position of 20 joints relative to the origin. Variability of subjects position and orientation toward the camera, variation of performance among different patients and noticeable difference in speed and duration of actions are the main challenges of this dataset. Humanobject occlusions and out of field-of-view body parts make the sensor unable to recover all of the body parts and add up to the challenges faced in this dataset.

CAD-60 dataset:

Daily activities rarely occur in controlled laboratory environment. It motivated researchers in Cornell University to create the *CAD-60 dataset* [54] for actions occurring in real environments. 4 subjects performed 12 different actions in 5 different environments where Depth, RGB, and skeleton data for each instance are captured by Kinect v1 sensor in 30 fps. Each action is performed at least one time by each subject. In total, dataset includes 60 sequences with average length of 45 seconds for each action. Skeleton data for each frame is presented by Euclidean position of 15 joints by taking sensor coordinates as the reference point. Insufficient training data and variable background are the main challenges of this dataset. The action instances are with different laterality as one of the subjects is left-handed. In order to compensate laterality, some of the proposed methods [43,52,54] also add a mirrored version of these instances to training data to achieve invariance toward handedness of the subjects.

UTD-MHAD dataset:

UTD-MHAD [9] is a multi-modal dataset which is released by University of Texas for multimodal activity recognition. The data was captured by Kinect v2 at 30 fps and a wearable inertial sensor. Four data modality including RGB, Depth, Skeleton, and inertial signal were registered in temporal synchronized mode using these sensors. The dataset includes 27 action. These actions are performed by 8 subjects (4 men and 4 women) in an environment with a fixed background. Every subject performed each action for 4 times. By removing three corrupted sequences, including start to end frame of a single action, the overall dataset includes 861 action instances. The skeleton data for each frame is presented by Euclidean position of 20 joints with respect to the sensor coordinates. In another taxonomy, this dataset categorizes actions in four sub-categories: Sport actions (e.g., bowling, tennis serve, and baseball swing), hand gestures (e.g., draw x, draw triangle, and draw circle), daily activities (knock on door, sit to stand, and stand to sit), and training exercises (e.g., arm curl, lunge, and squat).

MSR Action 3D Dataset:

MSR Action 3D Dataset [33] is the first public RGB-D action dataset which is created by Microsoft Research Redmond in cooperation with the University of Wollongong in 2010. The dataset is recorded by Kinect v1 in 15 fps, and includes 20 actions involving different body parts. Each action is performed by 10 subjects for 2-3 times. In total 567 sequences exist in the dataset which their lengths vary between 13 and 67 frames. Each sequence includes an action which is manually segmented. The dataset also included depth and skeleton data of each action. Skeleton in each frame represented by Euclidean position of 20 joints relative to the origin which is the sensor coordinates. In all instances, subjects perform actions with a fixed position facing towards the camera with a controlled background.

Table 1: Summary of datasets

Dataset Name	Actions	Subjects	Sequences	Joints	Year
UTKinect-Action [69]	10	10	199	20	2012
CAD-60 [54]	12	4	60	15	2011
UTD-MHAD [9]	27	8	861	20	2015
MSRAction3D [33]	20	10	557	20	2010

A summary of general characteristics of the four datasets used in our experiments for evaluating the proposed method is shown in Table 1.

Experimental Settings: In our experiments, 3D coordinates of the skeleton joints are converted from world coordinates into subject coordinates by taking Hip Center joint as coordinate systems origin in each frame. The obtained results in each dataset is compared to the methods that use only skeleton data for recognition tasks. The three input parameters of our framework are individually tuned for each dataset. The first parameter in Equation 1is the temporal offset (k') which is used for constructing temporal differ-

Table 2: Investigate Parameters of Approach

Dataset Name	Investigated Interval & steps		
	Temporal offset	Cluster numbers	Neuron numbers
UTKinect-Action [69]	4:1:20	150:10:250	500:100:3500
CAD-60 [54]	10:10:150	100:10:250	500:100:3500
UTD-MHAD [9]	4:1:20	150:10:250	500:100:3500
MSRAction3D [33]	4:1:20	150:10:250	500:100:3500

ence (Xtd) in the feature vector. The second parameter is the number of clusters in *K-means* clustering method which is used to extract key poses from all of the training poses. In other words, it is the number of key poses. The last parameter that needs tuning is number of neurons in the hidden layer of the *ELM* which is used for classification. We start with big steps and wider range of parameters for the framework and narrow down the intervals to find the optimal values. As it is shown in Table2, we empirically determined the optimal intervals and best fit of step size which ensure the best overall performance of the recognition framework. We perform a random initialization of the cluster centers in *K-means* method to calculate the key poses. Therefore, the proposed method is repeated 20 times on each dataset and then the best result among them is reported and compared with the state-of-the-art approaches.

In the seminal work based on UTKinect-Action dataset [69], the authors used *leave-one-sequence-out cross-validation (LOSeqO)* protocol for their evaluations. In this protocol they randomly select one sequence at a time from the entire dataset as test instance and use the remaining sequences as training data. This process is repeated certain times and average of the obtained results are used as the final performance [76]. In our experiment, we follow the Cross-subject protocol in [59]. Subjects 1,3,5,7 and 9 are selected for training and subjects 2,4,6,8 and 10 for testing. This evaluation protocol is more realistic since the test subject‘s actions are kept completely out of the training set. We used Table 2 to find the optimized parameters for *UTKinect-Action dataset*. We obtained the best performance by setting temporal offset to **6**, key pose number to **160** and number of neurons to **3100**. The results and comparison with the state-of-the-art methods are shown in Table3.

As shown in Table3, as far as we know, the best performance overall skeleton-based approaches on *UTKinect Action Dataset* is obtained by our method. Performance accuracy analysis of our method on this dataset based on the confusion matrix (Figure4) shows that in **10%** of the test samples of *"push"* action are misclassified as *"throw"* action. Similarity between poses of these two action and nose in skeleton joint positions are potentially the main causes of this recognition failure. Our method recognizes nine out of ten actions by **100%** accuracy.

Sung, J., et al. [54] presented two types of protocol called *"New Person"* and *"Have Seen"* for evaluating *CAD-60 dataset*.

Table 3: Comparison with the state-of-the-art results on *UTKinect-Action*dataset

Feature Engineering	Method	Accuracy (%)
Hand-crafted	HOJ3D [69] (LOSeqO)	90.92
	Lie Group [59]	97.8
	Spatiotemporal SHs [73]	93.0
	Our method	**98.98**
Learned representations	RDF-based [40]	92.0
	Max-Margin Multitask [71] (LOOCV)	98.8
	LMNN [38](LOOCV)	98.0
RNN-LSTM	Multilayer LSTM [77]	95.96
	ST-LSTM [35]	95.0

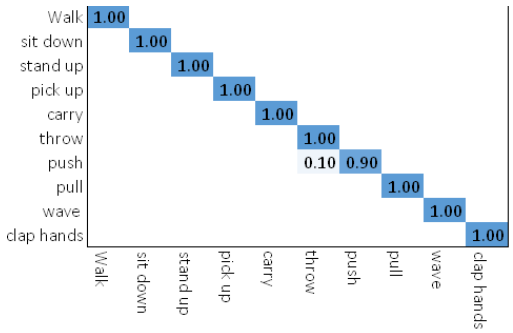

Fig. 4: Confusion matrix of UTKinect dataset

They used precision/recall measures to evaluate their proposed method. In our experiment we adopt *"New Person"* protocol for evaluations. This protocol is defined as a *Leave-one-subject-out cross-validation*. Therefore, one subject is used for test while the other 3 subjects are kept for training. In *CAD-60 dataset*, one of the four subjects is left-handed (subject 3). We use mirroring operations before constructing the feature vector in order to convert laterality of the actions and make it similar to right-handed data. [81] Achieved state-of-the-art results on *CAD-60 dataset*. In their approach subject number 2 is used for test and the other 3 subjects (1, 3 and 4) for training. We adopt the same setting in our experiments. Length of the actions in this dataset is longer than the previous dataset. Using Table2 we tried different parameter intervals and step sizes. By examining all possible scenarios for the parameters in these intervals, we obtained the best performance with value **50** for temporal offset, **210** for key poses number and **3100** for number of neurons on *CAD-60 Dataset*.

The performance of our method on *CAD-60 Dataset* is shown in Table4 shows performance of our approach and comparison with successful approaches in the literature on *CAD-60*. It can be noticed from Table4. That our proposed method achieves competitive performance compared to Hand-crafted skeleton-based methods. Except subject 3, all of the actions in different environments **3** are recognized with **100%** success. As it is clear from the confusion matrix (Figure5), recognizing *"talking on coach"* instead of *"relaxing on coach"* is the single failure that occurs in subject threes instances. Insufficient training sample is the main reason for this failure. Since there is only one test instance available for *"relaxing on coach"* for this subject, the calculated precision turns out to be the undefined value of $0/0$. To compute av-

Table 4: Comparison with the state-of-the-art results on *CAD-60* dataset *notice that in this method they use both depth and skeleton informatione

Feature Engineering	Method	Precision (%)	Recall (%)
Hand-crafted	MEMM [54]	67.9	55.5
	3D posture [18]	77.3	76.7
	Pose Kinetic Energy [52]	93.8	94.5
	Decision-Level Fusion [81]*	96.4	84.6
	Our method	**98.5**	**99**
Learned representations	M-L codebooks of key pose [79]	97.4	95.8
	Self-organizing neural int [43]	91.9	90.2
	RF-Key pose [42] (Random + Still)	81.8	80.0

erage precision of actions in the *"living room"* environment we considered this value as zero.

The common practice in *UTD-MHAD dataset* [9] is to perform *cross-subject evaluation* protocol which was suggested by its providers. In this protocol half of the subjects (1, 3, 5 and 7) are taken for training and the other half (subjects 2, 4, 6 and 8) for testing. In our experiment we used this setting for evaluating our proposed method too. Similar to previous datasets, we investigates the optimal parameters by going through values indicated in Table2. We obtained

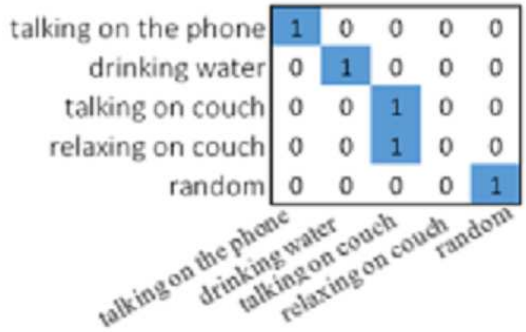

Fig. 5: Confusion matrix of Livingroom Actions (subject 3)

Table 5: Comparison with the state-of-the-art results on *UTD-MHAD* *notice that in this method they use data other than skeleton information

Feature Engineering	Method	Accuracy (%)
Hand-crafted	Kinect&Inertial [9]	79.1
	Kinect&Inertial fusion [10]	91.5
	Cov3DJ [22]	85.5
	Our method	**90.7**
Learned representations	ELC-KSVD [22]	76.1
CNN	SOS_ based CNN [22]	86.9
	JTM_ CNN [68]	85.8

the best performance with value **8** for temporal offset, **130** for key poses number and **3100** for number of neurons on *UTD-MHAD Dataset*. As shown in Table5 to the best of our knowledge, the best performance among all skeleton-based approaches on *UTD-MHAD Dataset*, is obtained by our method.

Analysis of confusion matrix of our method on this dataset (Figure6) showed that actions sharing common poses lead to inaccurate recognition. For instance *"draw circle (counter clockwise)"* is classified with **44%** accuracy while in **25%** of samples it is misclassified as *"draw circle (clockwise)"*. In a similar situation, *"Clap"* is misclassified **31%** of time as *"cross arms in the chest"* action. Nevertheless, **13** actions out of **27** are recognized with **100%** accuracy. Having highly distinctive poses leads the framework to distinguish these actions with perfect accuracy. There are two settings which have been used in previous studies to evaluate *MSR-Action3D* [33] dataset. The first one was proposed in the seminal paper [33] of *MSR-Action3D* dataset where all of the actions in the dataset are divided into three subcategories *(AS1, AS2 and AS3)* showed in Table6. Every subcategory consists of 8 action classes which training and classification of actions are performed independently on each category. In sub categories *AS1 and AS2* actions with similar motion are grouped together. These categories are used for evaluating distinctive ability of algorithms for recognizing actions with similar structure. Sub category *AS3* contains actions consist of complex body dynamics and is used for evaluation of diversity of a method. The overall performance of a system is obtained by averaging the performance of sub categories.

The second experimental protocol which is suggested in [67] keeps all of the **20** actions in a single set for training and testing without splitting the dataset. This makes the classification even harder compared to the first setting. In our experiments we use the first setting with cross-subject protocol similar to [59]. We use half of the subjects (1, 3, 5, 7 and 9) for training and the other half (2, 4, 6, 8 and 10) for testing. By examining all of the possible scenarios for the parameters indicated in Table2, we obtained the best performance when we set the temporal offset to **8**, number of key poses to **130** and number of neurons to **3100** on *MSR Action 3D* Dataset. The performance of our method on *MSR Action 3D* Dataset and comparison with skeleton-based state-of-the-art methods are shown in Table7. Depending on the feature type, the methods are categorized into hand-crafted or automatic types.

Table 6: Three action subset of *MSR Action 3D*

AS1	AS2	AS3
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

Our proposed method achieves acceptable performance among Hand-crafted methods when features are calculated only in Euclidean space without transformation into another

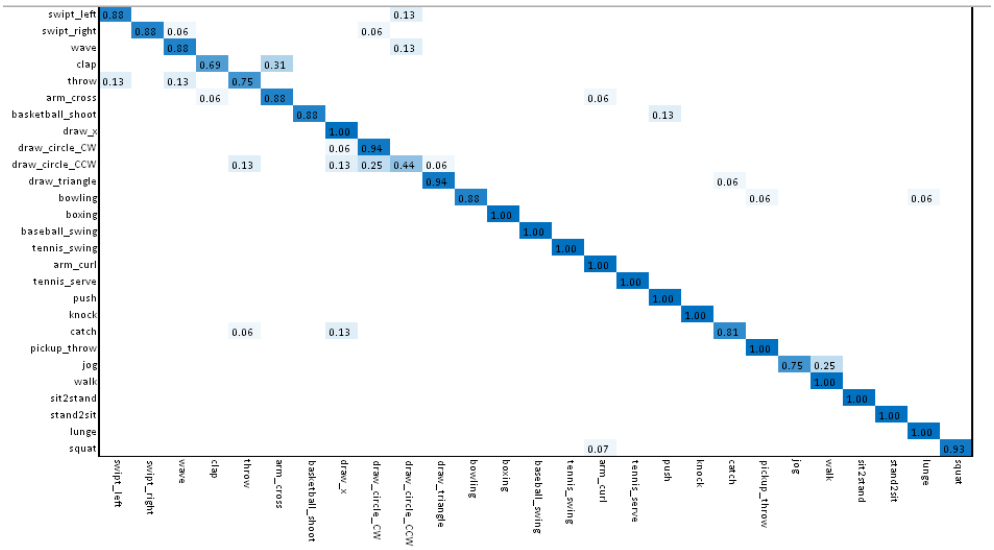

Fig. 6: Confusion matrix of *UTD-MHAD dataset*

Table 7: Comparison with the state-of-the-art results on *MSR Action3D*

Feature Engineering	Method	Accuracy(%) AS1	AS2	AS3	Average
Hand-crafted	Pose-based [62]	-	-	-	90.2
	Pose-based [62]	-	-	-	90.2
	HOJ3D [69]	-	-	-	78.9
	Lie Group [59]	95.3	83.9	98.2	92.5
	Spatiotemporal SHs [73]	89.7	91.7	92.5	90.9
	Our method	**94.3**	**88.4**	**97.4**	**93.3**
Learned representations	LMNN [38]	-	-	-	97.1
	Trajectory let [47]	96.4	97.5	100	97.9
	Moving Pose lets [55]	89.8	93.5	97.0	93.5
	Max-Margin Multitask [77]	-	-	-	95.62
RNN	HBRNN-L [14]	93.3	94.6	95.5	94.5

space like [59,73]. The approaches such as [38] that employ data mining techniques to select distinctive features were achieved superior results in comparison. However, performance improvement in action recognition in these methods coincides with an increase in computational cost especially in the training phase. As shown in Table7 our methods generated relatively better result compared to [14,55,73] on *AS3* which contain actions with complex structure. Compared to other two sub categories, actions in sub category *AS2* are more challenging for our framework and results in less accuracy in performance due to complexity of the actions. It can be clearly seen from the confusion matrix (Figure7) that the *"hand catch"* action is correctly classified only in **50%** of the test samples. However, this action misclassified in **17%** of samples as *"Draw x"* and other **17%** samples as *"Forward kick"* action. In *AS1* sub category the highest misclassification rate happens in *"Pickup throw"* action, where it misclassified in **14%** of the samples as *"Bend"*, in **7%** as *"Hammer"* and **7%** as *"Hight throw"* actions. Lack of

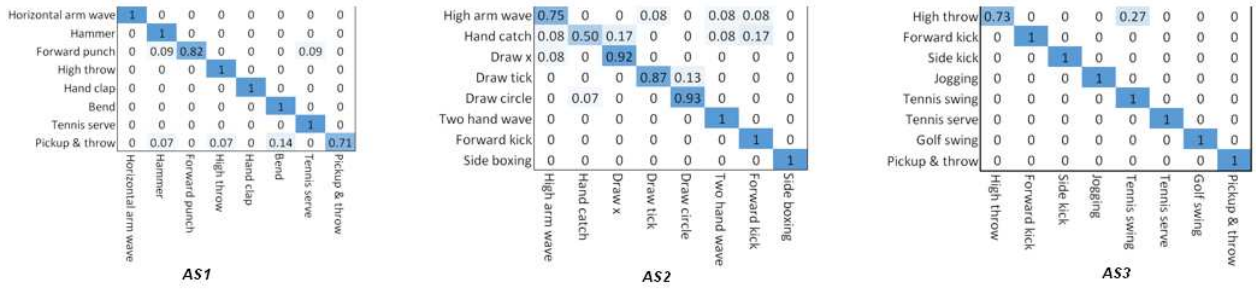

Fig. 7: Confusion Matrixes of MSR Action3D dataset

producing distinictive key poses for each action class is the main reason for recognation failur. For example in case of *"Pickup throw"* action, our approach generates same key poses with different temporal orders compared to the two other confusing actions. Even though the generated poses comprise time information, during complex action encoding procedure, the framework loses the temporal order of poses in the sequence for some of the actions. Based on our experiments small-sized codebook does not generate sufficiently diverse code words to discriminate all of the actions and the one with a large size is highly prone to noise. Most of the key-pose based methods usually use *HMMs* to define the action and model temporality, hence, number of the generated key-poses are limited. One of the main privileges of our method to these key-pose based methods is that rather than generating the action sequence using the key-poses, we find the available key-poses in the actions using a dictionary populated with sufficient key-poses where absence of a key-pose is still a valuable information. However, sometime higher number of key-poses add up to the noise in recognitions. Tuning the number of the key-poses is an important aspect that have a great impact on robustness of recognition in our method and needs to be carefully done.

5 Conclusion and future work

In this study, we proposed a novel *bag-of-poses* framework for 3D action recognition based on a set of predefined spatio-temporal poses. Most of the studies available in the literature regarding pose-based action recognition have used *generative* or *bag-of-poses* approaches. The main disadvantages of the generative methods are the exceeding need for training data and challenging parameter tuning which is usually performed manually. Accordingly, the main drawback of *bag-of-poses* approaches is to not to consider the concept of time among the poses when trying to encode an action. As a solution, our main objective is to improve the *bag-of-poses* approach by embedding temporal information using the key-pose descriptors. The proposed descriptor enables us to distinguish between two poses with the same skeleton configuration while different temporal order exists in

an action sequence. The pose descriptor is extracted from Euclidean coordinates of the skeleton joints without transforming coordinates to another space. The suggested framework is validated with four publicly available benchmark 3D action datasets, and produced state-of-the-art results on the three datasets and competitive result on the fourth dataset. Our method can be enhanced in order to obtain more accurate results. The main aspect that needs to be improved is to recognize interactive actions between subjects. This is mainly because the framework does not benefit from the context information and interaction with the objects in the environment. As a future study, we will investigate on this subject to improve the results by utilizing depth and contextual information.

References

1. Aggarwal, J., Xia, L.: Human activity recognition from 3d data: A review. Pattern Recognition Letters **48**, 70–80 (2014)
2. Aggarwal, J.K., Ryoo, M.S.: Human activity analysis: A review. ACM Computing Surveys (CSUR) **43**(3), 16 (2011)
3. Amor, B.B., Su, J., Srivastava, A.: Action recognition using rate-invariant analysis of skeletal shape trajectories. IEEE transactions on pattern analysis and machine intelligence **38**(1), 1–13 (2016)
4. Ben-Arie, J., Wang, Z., Pandit, P., Rajaram, S.: Human activity recognition using multidimensional indexing. IEEE Transactions on Pattern Analysis and Machine Intelligence **24**(8), 1091–1104 (2002)
5. Bobick, A.F., Davis, J.W.: The recognition of human movement using temporal templates. IEEE Transactions on pattern analysis and machine intelligence **23**(3), 257–267 (2001)
6. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2d pose estimation using part affinity fields. arXiv preprint arXiv:1611.08050 (2016)
7. Chaaraoui, A.A., Padilla-Lpez, J.R., Climent-Prez, P., Flrez-Revuelta, F.: Evolutionary joint selection to improve human action recognition with rgb-d devices. Expert systems with applications **41**(3), 786–794 (2014)
8. Chang, C.C., Lin, C.J.: Libsvm: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST) **2**(3), 27 (2011)
9. Chen, C., Jafari, R., Kehtarnavaz, N.: Utd-mhad: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor. In: Image Processing (ICIP), 2015 IEEE International Conference on, pp. 168–172. IEEE
10. Chen, C., Jafari, R., Kehtarnavaz, N.: A real-time human action recognition system using depth and inertial sensor fusion. IEEE Sensors Journal **16**(3), 773–781 (2016)
11. Chen, X., Koskela, M.: Skeleton-based action recognition with extreme learning machines. Neurocomputing **149**, 387–396 (2015)
12. Chron, G., Laptev, I., Schmid, C.: P-cnn: Pose-based cnn features for action recognition. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3218–3226
13. Debes, C., Merentitis, A., Sukhanov, S., Niessen, M., Frangiadakis, N., Bauer, A.: Monitoring activities of daily living in smart homes: Understanding human behavior. IEEE Signal Processing Magazine **33**(2), 81–94 (2016)
14. Du, Y., Wang, W., Wang, L.: Hierarchical recurrent neural network for skeleton based action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1110–1118
15. Efros, A.A., Berg, A.C., Mori, G., Malik, J.: Recognizing action at a distance. In: ICCV, vol. 3, pp. 726–733
16. Eweiwi, A., Cheema, M.S., Bauckhage, C., Gall, J.: Efficient pose-based action recognition. In: Asian Conference on Computer

Vision, pp. 428–443. Springer

17. Fothergill, S., Mentis, H., Kohli, P., Nowozin, S.: Instructing people for training gestural interactive systems. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1737–1746. ACM
18. Gaglio, S., Re, G.L., Morana, M.: Human activity recognition process using 3-d posture data. Human-Machine Systems, IEEE Transactions on **45**(5), 586–597 (2015)
19. Gong, W., Bagdanov, A.D., Roca, F.X., Gonzlez, J.: Automatic key pose selection for 3d human action recognition. In: International Conference on Articulated Motion and Deformable Objects, pp. 290–299. Springer
20. Han, F., Reily, B., Hoff, W., Zhang, H.: Space-time representation of people based on 3d skeletal data: A review. Computer Vision and Image Understanding (2017)
21. Han, J., Shao, L., Xu, D., Shotton, J.: Enhanced computer vision with microsoft kinect sensor: A review. IEEE transactions on cybernetics **43**(5), 1318–1334 (2013)
22. Hou, Y., Li, Z., Wang, P., Li, W.: Skeleton optical spectra based action recognition using convolutional neural networks. IEEE Transactions on Circuits and Systems for Video Technology (2016)
23. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme learning machine: theory and applications. Neurocomputing **70**(1), 489–501 (2006)
24. Hussein, M.E., Torki, M., Gowayyed, M.A., El-Saban, M.: Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. In: Twenty-Third International Joint Conference on Artificial Intelligence
25. Johansson, G.: Visual motion perception. Scientific American (1975)
26. Kapsouras, I., Nikolaidis, N.: Action recognition on motion capture data using a dynemes and forward differences representation. Journal of Visual Communication and Image Representation **25**(6), 1432–1445 (2014)
27. Klaser, A., Marszaek, M., Schmid, C.: A spatio-temporal descriptor based on 3d-gradients. In: BMVC 2008-19th British Machine Vision Conference, pp. 275: 1–10. British Machine Vision Association
28. Kovashka, A., Grauman, K.: Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2046–2053. DOI 10.1109/CVPR.2010.5539881
29. Kuehne, H., Arslan, A., Serre, T.: The language of actions: Recovering the syntax and semantics of goal-directed human activities. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 780–787
30. Laptev, I., Lindeberg, T.: Space-time interest points. In: Proceedings Ninth IEEE International Conference on Computer Vision, pp. 432–439 vol.1. DOI 10.1109/ICCV.2003.1238378
31. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, pp. 1–8. IEEE
32. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
33. Li, W., Zhang, Z., Liu, Z.: Action recognition based on a bag of 3d points. In: Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on, pp. 9–14. IEEE
34. Lillo, I., Niebles, J.C., Soto, A.: Sparse composition of body poses and atomic actions for human activity recognition in rgb-d videos. Image and Vision Computing **59**, 63–75 (2017)
35. Liu, J., Shahroudy, A., Xu, D., Wang, G.: Spatio-temporal lstm with trust gates for 3d human action recognition. In: European Conference on Computer Vision, pp. 816–833. Springer
36. Lu, G., Zhou, Y., Li, X., Kudo, M.: Efficient action recognition via local position offset of 3d skeletal body joints. Multimedia Tools

and Applications **75**(6), 3479–3494 (2016)

37. Luo, J., Wang, W., Qi, H.: Group sparsity and geometry constrained dictionary learning for action recognition from depth maps. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1809–1816
38. Luvizon, D.C., Tabia, H., Picard, D.: Learning features combination for human action recognition from skeleton sequences. Pattern Recognition Letters (2017)
39. Minhas, R., Baradarani, A., Seifzadeh, S., Wu, Q.J.: Human action recognition using extreme learning machine based on visual vocabularies. Neurocomputing **73**(10), 1906–1917 (2010)
40. Negin, F., Akgl, C.B., Yksel, K.A., Eril, A.: An rdf-based action recognition framework with feature selection capability, considering therapy exercises utilizing depth cameras. Journal of Theoretical and Applied Computer Science **8**(3), 3–22 (2014)
41. Negin, F., zdemir, F., Akgl, C.B., Yksel, K.A., Eril, A.: A decision forest based feature selection framework for action recognition from rgb-depth cameras. In: International Conference Image Analysis and Recognition, pp. 648–657. Springer
42. Nunes, U.M., Faria, D.R., Peixoto, P.: A human activity recognition framework using max-min features and key poses with differential evolution random forests classifier. Pattern Recognition Letters (2017)
43. Parisi, G.I., Weber, C., Wermter, S.: Self-organizing neural integration of pose-motion features for human action recognition. Frontiers in neurorobotics **9**, 3 (2015)
44. Peng, X., Wang, L., Wang, X., Qiao, Y.: Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice. Computer Vision and Image Understanding **150**, 109–125 (2016)
45. Poppe, R.: A survey on vision-based human action recognition. Image and vision computing **28**(6), 976–990 (2010)
46. Presti, L.L., La Cascia, M.: 3d skeleton-based human action classification: a survey. Pattern Recognition **53**, 130–147 (2016)
47. Qiao, R., Liu, L., Shen, C., van den Hengel, A.: Learning discriminative trajectorylet detector sets for accurate skeleton-based action recognition. Pattern Recognition (2017)
48. Ramanathan, M., Yau, W.Y., Teoh, E.K.: Human action recognition with video data: research and evaluation challenges. IEEE Transactions on Human-Machine Systems **44**(5), 650–663 (2014)
49. Sadanand, S., Corso, J.J.: Action bank: A high-level representation of activity in video. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pp. 1234–1241. IEEE
50. Scovanner, P., Ali, S., Shah, M.: A 3-dimensional sift descriptor and its application to action recognition. In: Proceedings of the 15th ACM international conference on Multimedia, pp. 357–360. ACM
51. Sempena, S., Maulidevi, N.U., Aryan, P.R.: Human action recognition using dynamic time warping. In: Electrical Engineering and Informatics (ICEEI), 2011 International Conference on, pp. 1–5. IEEE
52. Shan, J., Akella, S.: 3d human action segmentation and recognition using pose kinetic energy. In: Advanced Robotics and its Social Impacts (ARSO), 2014 IEEE Workshop on, pp. 69–75. IEEE
53. Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., Moore, R.: Real-time human pose recognition in parts from single depth images. Communications of the ACM **56**(1), 116–124 (2013)
54. Sung, J., Ponce, C., Selman, B., Saxena, A.: Unstructured human activity detection from rgbd images. In: Robotics and Automation (ICRA), 2012 IEEE International Conference on, pp. 842–849. IEEE
55. Tao, L., Vidal, R.: Moving poselets: A discriminative and interpretable skeletal motion representation for action recognition. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 61–69
56. Tran, D., Torresani, L.: Exmoves: Mid-level features for efficient action recognition and video analysis. International Journal of

Computer Vision **119**(3), 239–253 (2016)

57. Varol, G., Salah, A.A.: Efficient large-scale action recognition in videos using extreme learning machines. Expert Systems with Applications **42**(21), 8274–8282 (2015)
58. Veeriah, V., Zhuang, N., Qi, G.J.: Differential recurrent neural networks for action recognition. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4041–4049
59. Vemulapalli, R., Arrate, F., Chellappa, R.: Human action recognition by representing 3d skeletons as points in a lie group. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 588–595
60. Vemulapalli, R., Arrate, F., Chellappa, R.: R3dg features: Relative 3d geometry-based skeletal representations for human action recognition. Computer Vision and Image Understanding **152**, 155–166 (2016)
61. Vishwakarma, S., Agrawal, A.: A survey on activity recognition and behavior understanding in video surveillance. The Visual Computer **29**(10), 983–1009 (2013)
62. Wang, C., Wang, Y., Yuille, A.L.: An approach to pose-based action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 915–922
63. Wang, C., Wang, Y., Yuille, A.L.: Mining 3d key-pose-motifs for action recognition. In: Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on, pp. 2639–2647. IEEE
64. Wang, H., Klser, A., Schmid, C., Liu, C.L.: Action recognition by dense trajectories. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, pp. 3169–3176. IEEE
65. Wang, H., Klser, A., Schmid, C., Liu, C.L.: Dense trajectories and motion boundary descriptors for action recognition. International journal of computer vision **103**(1), 60–79 (2013)
66. Wang, J., Liu, Z., Wu, Y.: Learning actionlet ensemble for 3D human action recognition, pp. 11–40. Springer (2014)
67. Wang, J., Liu, Z., Wu, Y., Yuan, J.: Mining actionlet ensemble for action recognition with depth cameras. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pp. 1290–1297. IEEE
68. Wang, P., Li, Z., Hou, Y., Li, W.: Action recognition based on joint trajectory maps using convolutional neural networks. In: Proceedings of the 2016 ACM on Multimedia Conference, pp. 102–106. ACM
69. Xia, L., Chen, C.C., Aggarwal, J.: View invariant human action recognition using histograms of 3d joints. In: Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on, pp. 20–27. IEEE
70. Yang, X., Zhang, C., Tian, Y.: Recognizing actions using depth motion maps-based histograms of oriented gradients. In: Proceedings of the 20th ACM international conference on Multimedia, pp. 1057–1060. ACM
71. Yang, Y., Deng, C., Tao, D., Zhang, S., Liu, W., Gao, X.: Latent max-margin multitask learning with skelets for 3-d action recognition. IEEE transactions on cybernetics **47**(2), 439–448 (2017)
72. Yao, A., Gall, J., Fanelli, G., Van Gool, L.: Does human action recognition benefit from pose estimation?". In: Proceedings of the 22nd British machine vision conference-BMVC 2011
73. Youssef, C.: Spatiotemporal representation of 3d skeleton joints-based action recognition using modified spherical harmonics. Pattern Recognition Letters **83**, 32–41 (2016)
74. Zanfir, M., Leordeanu, M., Sminchisescu, C.: The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2752–2759
75. Zelnik-Manor, L., Irani, M.: Event-based analysis of video. In: Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, vol. 2, pp. II–II. IEEE
76. Zhang, J., Li, W., Ogunbona, P.O., Wang, P., Tang, C.: Rgb-d-based action recognition datasets: A survey. Pattern Recognition **60**, 86–105 (2016)

77. Zhang, S., Liu, X., Xiao, J.: On geometric features for skeleton-based action recognition using multilayer lstm networks. In: Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on, pp. 148–157. IEEE

78. Zhu, F., Shao, L., Xie, J., Fang, Y.: From handcrafted to learned representations for human action recognition: a survey. Image and Vision Computing **55**, 42–52 (2016)

79. Zhu, G., Zhang, L., Shen, P., Song, J.: Human action recognition using multi-layer codebooks of key poses and atomic motions. Signal Processing: Image Communication **42**, 19–30 (2016)

80. Zhu, W., Lan, C., Xing, J., Zeng, W., Li, Y., Shen, L., Xie, X.: Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks. arXiv preprint arXiv:1603.07772 (2016)

81. Zhu, Y., Chen, W., Guo, G.: Fusing multiple features for depth-based action recognition. ACM Transactions on Intelligent Systems and Technology (TIST) **6**(2), 18 (2015)