Dynamic Time Warping-Based Features with Class-Specific Joint Importance Maps for Action Recognition Using Kinect Depth Sensor

Hoda Mohammadzade*, Soheil Hosseini, Mohammad Reza Rezaei Dastjerdehei, and Mohsen Tabejamaat

***Abstract*— This paper proposes a novel 3D action recognition technique that uses time-series information extracted from depth image sequences for use in systems of human daily activity monitoring. To this end, each action is represented as a multi-dimensional time series, where each dimension represents the position variation of one skeleton joint over time. The time series is then mapped onto a vector space using Dynamic Time Warping (DTW) distance. Furthermore, to employ the correlation-distinctiveness relationship of the sequences in recognition, this vector space is remapped onto a discriminative space using the regularized Fisher method, where final decisions about the actions are made. Unlike other available methods, the time-warping used in the mapping strategy makes the feature space robust to temporal variations of the motion sequences. Moreover, our method eliminates the need for a complicated design method for extracting the static and dynamic information of a motion sequence. Furthermore, most existing methods treat all skeletal joints identically for different actions, while some joints are more discriminative to distinguish a specific action. Thanks to the nature of the proposed features, we propose to use a separate set of discriminative joints, called joint importance map for each class of action. Evaluation results on four well-known datasets, TST, UTKinect, UCFKinect, and NTU RGB+D show competitive performance with the state-of-the-art methods in human action recognition.**

***Keywords*: 3D action recognition, time series, dynamic time warping, feature space, joint importance map.**

I. Introduction

HUMAN action recognition is one of the critical research areas in machine vision, which has many potential applications in various fields, including health, security, and human-machine interaction. Among the various applications in these fields, one could mention the monitoring of sick and older people, surveillance in public places, and entertainment. In general, in terms of complexity, we can classify an activity into one of three levels of gesture, action, and behavior, and this article focuses on the middle level, i.e., action, which is a sequence of several seconds of a few gestures. For example, activities such as sitting, picking an object, and falling on the ground are referred to as actions.

Monitoring devices consist of two types: wearable and non-wearable. Both types have been utilized to recognize the different actions and movements. Wearable sensors have become popular in many applications such as medical, entertainment, security, and commercial fields. Despite the fact that using wearable is not always convenient, the measurements using these devices, which are mostly developed as smartwatches or smartbands, can only take place on the location where the device is worn such as the upper limb. Therefore, using non-wearable devices to track body movement is more preferable. Different non-wearable devices have been adopted for action recognition including camera [1] and radar-based [2] systems. Despite the advantages offered by these devices, some hindrances can emerge: RGB cameras are susceptible to problems such as background cluttering, view angle and privacy threats. One the other hand, radar-based sensors do not face the same limitations and have high penetration ability which can detect even through-wall human movements [3], however, their action recognition performance is yet to be improved [4]. In recent years, the emergence and commercialization of Kinect depth mapping sensors, by removing many of the above mentioned problems, has created the ground for advancement in methods for action recognition using three-dimensional data. Following the important work in [5] to extract joint positions using depth images, methods for action recognition using skeletal information became of great interest. Compared to the depth images, skeletal models provide more compact and efficient information and eliminate disadvantages such as occlusion of body parts by each other and the effect of body volume in the recognition process. By solving the problem of data acquisition, the only remaining challenge in identifying an action is how to model a motion sequence, which has become a focus of attention of researchers in recent years. The methods presented in this field are mainly divided into four categories. In the first category, motion is modeled as a single or a set of time series, and recognized by one-to-one

Hoda Mohammadzade is with the Electrical Engineering Department, Sharif University of Technology, Tehran, Iran 11155-8639 (e-mail: hoda@sharif.edu).

Soheil Hosseini was with the Electrical Engineering Department, Sharif University of Technology, Tehran, Iran 11155-8639. He is now with the Department of Biomedical Engineering at University of Iowa (e-mail: solliv@gmail.com).

Mohammad Reza Rezaei_Dastjerdehei is with the Electrical Engineering Department, Sharif University of Technology, Tehran, Iran 11155-8639 (e-mail: rezaeidastjerdehei.mohammadreza@ee.sharif.edu).

Mohsen Tabejamaat was with the Electrical Engineering Department, Sharif University of Technology, Tehran, Iran 11155-8639 (e-mail: m.tabejamaat@sharif.edu).

* corresponding author

matching of these series with those extracted from training data [6-8]. In the second category, the action is defined as a set of key skeletal states (skeletal gestures) and encoded as a sequence of indices corresponding to the key states [9-12]. The universal principle of the methods in the third category is the exploitation of manifold properties in the motion sequences distribution space [13, 14]. These methods try to provide a clearer picture of the levels of similarity in comparing motion actions, using the curvature property in a non-Euclidean space. Finally, the distinctive feature of methods in the fourth category is the use of mostly deep convolutional-recurrent neural networks in modeling the motion process [15, 16]. In this midst, the desirable method is clearly one that can satisfy the following conditions simultaneously: (a) has the least dependence on data preprocessing, (b) the computational complexity of its feature extraction phase is minimal, (c) considers the spatial and temporal changes in motion sequences simultaneously in the recognition process, (d) accounts for the temporal variations in the motion sequences, and (e) incorporates the correlation-distinctiveness relationship of the sequences in the decision-making process. Unfortunately, present methods focus only on one or more of the above conditions and do not fulfill all conditions simultaneously. For example, in the first category, the correlation-distinctiveness relationship of sequences is ignored, which deteriorates feature discrimination capability. In the second category, how to identify keyframes of a sequence and its sensitivity to system free parameters, pose a fundamental challenge. The underlying problem in the third category is defining a proper criterion for comparison and evaluation of different sub-manifolds. Finally, in methods based on deep learning, the time-consuming training process, and the need for a significant number of training samples are the main challenges of the system.

Since different actions can differ in all or only some of the frames, the distribution of motion sequences follows a nonlinear structure, neglecting, which would reduce the recognition accuracy of the system. Methods presented to cope with this structure fall mainly under two general categories: (a) methods based on the manifold properties, and (b) methods based on the kernel trick. In the former, it is assumed that large-dimensional nonlinear data usually fits on smaller-dimensional nonlinear manifolds. Therefore, its classification will be better performed in the non-Euclidean space. Although how to compare data placed on manifolds is itself a significant challenge for these methods. In the latter, nonlinear data is mapped to a larger-dimensional space using the kernel trick, so that the nonlinear structure of data becomes linearly separable in this space. Although again, how to determine the kernel type and its parameters are fundamental and challenging issues. Besides, in existing kernel methods, it is necessary first to convert the time series to one-dimensional data, and then apply the kernel trick to the resulting vectors. This will cause a significant sensitivity in kernel methods to temporal variations in motion sequences. Therefore, a comparison of the test sample with the constructed vectors will only be possible after the construction of the features vector upon motion completion.

Dynamic Time Warping (DTW) is one of the standard methods of aligning time series, also used in human action recognition applications. Still, this method does not consider the nonlinear structure of the correlation-distinctiveness relationship in training sequences, and therefore its use in classifying motion sequences is not efficient.

Considered the above disadvantages, the current paper presents a method of action recognition using features based on DTW distance. To this purpose, the motion sequences are first translated into an optimal vector space using DTW distance. This effectively reduces the dimensions of the time series and simultaneously eliminates the effect of temporal variations in similar sequences. Also, the extracted features are such that they can be used to train the classifier and learn discriminant features. Since the proposed method uses only one of the sequences in each class as the class reference for calculating the feature vector, its computational complexity is greatly reduced. Furthermore, the regularized Fisher method is used to remap the vector space onto a discriminative space. This will cause the correlation-distinctiveness relationship of the sequences to be considered in the decision-making process. In addition, decisions are made in a much smaller dimension than the vector space dimension, which itself mitigates the curse of dimensionality present in the vector space.

Sections beyond here are organized as follows. Section (2) provides an overview of the works in human action recognition. Section (3) introduces the proposed method and its implementation. In Section (4), the simulation results of the algorithm and its comparison with other available methods are presented. Finally, Section (5) is dedicated to conclusions from the paper.

II. Related Works

This section provides a brief overview of prominent methods in action recognition based on human skeletal features. In [9], Lv et al. presented an action recognition system based on MoCap data that uses the distance between the joints as features in each frame and then uses a combination of the Hidden Markov Model (HMM) and the multi-class Adaboost algorithm to classify skeletal sequences. Guo et al. [8] presented a method for describing rotation and relative velocity (RRV) to describe trajectories of different body parts and used DTW to handle temporal variations in different RRVs. In [6-8], DTW distance has been used as a feature for nearest neighbor classification. In [17], the covariance matrix of the time series resulting from coordinate changes of the joints was extracted as motion features, and classified by a linear kernel Support Vector Machine (SVM). They also used a hierarchical method to extract the covariance-based features vector to incorporate the temporal order of movements in the recognition process. Lu et al.[18] used a bag-of-words framework to assemble position offset of 3D skeletal body joints to describe a motion sequence, along with the Bayes classification algorithm to classify the descriptors. The authors of [19] used a Histogram of Oriented Displacements (HOD) in projections of a skeleton model as features of skeletal frames. In [20], two descriptors of skeletal coordinates and skeletal angles sequences, obtained from derivatives of body parts trajectories, are used to describe human actions. Wang et al. [21] describe each action using the histogram of spatiotemporal indices of different moving body parts and use SVM to classify histogram vectors. Methods have been proposed to exploit only important joints in the recognition process, using different definitions of importance. In [22], the variance of inter-joint angles or their maximum angular velocity have been used to assess importance at each frame. However, such a frame-wise approach as opposed to combining information from 'all' frames to arrive at importance, could make comparison of joints in different actions difficult, as the important joints could change by the frame. In [23], differential entropy of the joint locations has been used to select informative joints, along a new idea of "skeleton contexts" to measure similarity between postures. In this method a bag of words scheme has been used for action recognition. The words are in fact quantized poses, and each action class includes a different sequence of these words. Modeling the sequences in different actions is performed using Conditional Random Fields (CRFs). Similar to our approach, joints most involved in an action in each class are used to create the model in this method, but differential entropy has rather been employed to select the joints. A recognition rate of 91.9% on the UTKinect dataset has been reported. Weng et al. [24] first divide each motion sequence into N parts, then calculate the distance between each joint descriptor in that part and each of the classes. Finally, the spatiotemporal matrix of these distances is calculated for each joint during the time frame of each part and is used as the time series descriptor of the corresponding action. In [10], each action is expressed as a sequence of key skeleton poses obtained by training a latent SVM. In this method, the pairwise relative positions of skeleton joints are used as a feature of the skeleton poses, which are mined with the aid of the latent SVM. Evangelidis et al. [16] coded the relative positions of joint quadruples using a multi-level representation of the Fisher vectors of these "quads" and used linear SVM to classify the extracted Fisher vectors. Vemulapalli et al. [13] introduced the rotations and translations of various skeleton sections in the 3D Euclidean space as points in a Lie group. Since a Lie group forms a curved manifold, a comparison of sequences placed on this manifold is made by features matching in the manifold tangent space. In [17], Yang et al. combine action information, including static posture, motion property, and overall dynamics, to form the features of an action sequence. Subsequently, by applying Principal Component Analysis (PCA) on the resulting composite vectors, they derive a set of discriminant features called EigenJoints and use the non-parametric Naïve Bayes Nearest Neighbor (NBNN) algorithm to classify the features. The authors of [7] use the histogram of the spherical distribution of joints as features of the action frames, and after applying Linear Discriminant Analysis (LDA), use feature clustering to encode the resulting features. Finally, HMM is used to model the action process. The success of deep convolutional-recurrent neural networks in video content recognition has also led to the widespread use of these networks in human action recognition using skeletal features. In this context, Du et al. [15] used a hierarchical recurrent Neural Network (RNN) to learn human skeletal states. In this method, the skeleton is divided into five parts, and a set of RNN subnets analyze local, mutual, and overall features of these parts simultaneously. In [16], a convolutional neural network has been used to extract joint coordinate co-occurrence features. As in [15], the authors of this article have tried to use a hierarchical approach to learning the features of joints. Accordingly, the joint changes feature is coded in the first, and the joints overlap feature is coded in the next layers of the network and then used.

Despite impressive performance on large-scale databases, Deep learning based strategies have still difficulties with modest amounts of training samples, where instead of hundreds or thousands, activities are mainly represented by a single or at most a few dozen training sequences. In addition to being time and effort consuming, doing such a huge data collection in no way matches the learning pattern of the brain in memorizing actions taking place in real world; for example, memorization of a complex sport movement (e.g. a martial arts move) while having just one observation, so accurate that we can replicate it in a largely flawless manner. We maintain that the learning strategy of our short-term memory is completely different from that currently employed by deep learning strategies, seeking to model activities using a considerable amount of training samples. We assert that while our brain needs a huge amount of training samples to initialize, it in no way requires such numbers to learn a new class of observations. Such drawbacks have shifted attention back towards more efficient more generalizable and less data hungry algorithms. Dynamic Time Warping (DTW) is one of such algorithms with an extraordinary power for supervised [14, 25-27], unsupervised [28], and weakly supervised [29] approaches for activity recognition. In [14], authors proposed to characterize each action as an element of Grassmann manifold along with using a DTW based tangent space measure to calculate pair distances of the manipulated trajectories. Switonski et al. [25] provided a detailed comparison for applying DTW distance measure on various feature representation and joint selection strategies. In [26], authors proposed to incorporate the multidimensionality characteristic of sequences in calculating DTW distance between trajectories. Choi et al. [26] modified the concept of vanilla DTW using a weighting strategy to incorporate the relevancy of body joints into the modeling of alignment procedure. Despite their success, these methods all suffer from a critical issue that is their constrained nature to either be used for calculating distances between trajectories or as aligning tools for matching sequences of different lengths. Unlike these algorithms, in this paper we provide a new insight into how we can use these warping strategies beyond just a distance or aligning measure as a feature extraction tool, which can be considered as a plug-and-play module in various applications of time series data characterization.

III. Dynamic Time Warping

Dynamic Time Warping [30] is a method for aligning two signals. The goal of this alignment is that two signals that have similar patterns of variations over time, but these variations happen at different rates or different times, will find a fairly complete match. To achieve this matching, each of the signals may undergo different local contractions or expansions. By means of a dynamic programming algorithm, this method provides a nonlinear correspondence between the time indices of the two signals, such that the sum of the distances between the two signal values at corresponding indices is minimized. Figure 1 shows an example of the alignment of two signals.

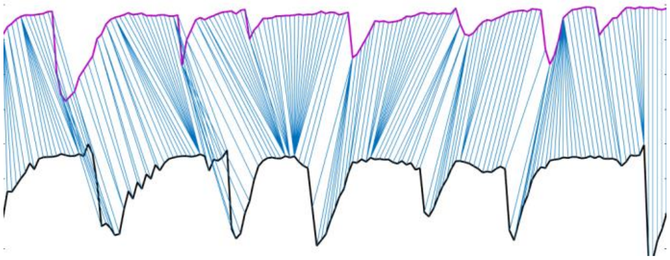


Fig. 1. Aligning two similar signals by the DTW method

DTW can be easily extended to multivariate time series. In this case, the cost to be minimized is the sum of the distances between the multi-dimensional values at corresponding indices. If X and Y are two p-dimensional time series of lengths M and N respectively, as

$$\mathrm{X}_{p\times M} = [\vec{x}_1, \dots, \vec{x}_M]$$
$$\mathrm{Y}_{p\times N} = [\vec{y}_1, \dots, \vec{y}_N] \qquad (1)$$

then DTW is looking for the warping path as pairs of indices $\{(i_1, j_1), \dots, (i_k, j_k), \dots, (i_K, j_K)\}$,such that conditions

$$\begin{gathered} i_1 = j_1 = 1 \\ i_K = M \\ j_K = N \qquad (2) \\ i_{k-1} \le i_k \quad , \quad 1 < k \le M \\ j_{k-1} \le j_k \quad , \quad 1 < k \le N \end{gathered}$$

hold for this path, and the cost function

$$Cost = \sum_{k=1}^{K} d(\vec{x}_{i_k}, \vec{y}_{j_k}) \qquad (3)$$

is minimized, in which d is a distance function for the vectors $\vec{x}_{i_k}$ and $\vec{y}_{j_k}$.

In attempts to use DTW for action recognition in the past [6-8], the nearest neighbor classifier is mainly used. In other words, past works have used only the DTW distance for recognition, and have not been able to extract features for training advanced classifiers. Extraction and use of these features in classifier training have the important advantage that the classifier can distinguish inter-class differences from intra-class dissimilarities of actions and thus perform recognition with higher precision. On the other hand, the nearest neighbor classifier, to perform optimally, needs to calculate the distance between the test sample with all the training samples, which, given the time-consuming nature of the DTW algorithm, introduces a heavy computational burden.

In this paper, we try to find a feature space in which, actions within classes lie close to each other, regardless of the temporal variations, and thus action classification is performed better.

IV. Extraction of DTW-Based Features

In the proposed method, each action is expressed as a set of time series or a multi-dimensional time series. This multi-dimensional time series contains the changes in the 3D coordinates of joints over time. Each dimension, in fact, represents the position of one of the three coordinates x, y, and z for each skeleton joint over time. For example, for a skeleton with 20 joints and an active length of 100 frames, the mentioned time series has 60 dimensions or sub-series, each of length 100.

The critical point and a challenge in action recognition are that not only different actions have different lengths (durations), but the same actions (for example, sitting) by different individuals (and even the same individual over different trials) have different lengths. In addition, different individuals may perform different parts of the same action at different rates. For example, in action 'throw,' one person may move the hand upwards more slowly than another person, but let go of the object more quickly. As a result, even sampling the actions to get equal lengths cannot provide a good correspondence between the frames. Therefore, common features like wavelets and those of frequency domain that relies on time correspondence of samples, cannot be effective in training classifiers, however powerful the classifiers be.

The purpose of this paper is to propose a method to use features in the DTW space for classifier training and thus make the algorithm robust to temporal variations in actions.

A. Inter-Action Features

In the proposed method, the DTW distance between the two-time series is used as a basis for feature extraction. DTW distance provides a nonlinear transformation between the original time series and a vector space in which action samples of same class lie close each other despite their temporal variations.

Let's represent the minimum cost in equation (2) as the DTW distance between the two time series X and Y, denoted by $DTW(\mathrm{X}, \mathrm{Y})$. We define the inter-action kernel features for the sample X as

$$\Phi_1(\mathrm{X}) = [DTW(\mathrm{X}, \Theta^1), \dots, DTW(\mathrm{X}, \Theta^G)]^T \qquad (4)$$

where $\Phi_1(.)$ is the inter-action feature generation function, Θ^g is a reference sample from the g-th class, and G is the number of classes. The method for finding reference samples is explained later. We use Euclidean distance for the function d in (3).

The inter-action features defined above, model the total distance between all joints between two action samples over time. However, they do not provide the specifics of the relative positions of individual joints in the two samples. In order to obtain more informative features, we define the distances between individual joints over time as the second group of inter-action features. To do this, we use the same alignment used in the previous step, i.e., the p-dimensional alignment (p is three times the number of joints), and consider the distance between two three-dimensional time series for each of the joints as the second group of features as

$$\Phi_2(\mathrm{X}) = \Bigg[\frac{1}{K_1}\Big(\sum_{k=1}^{K_1} d\big(\vec{x}_{1,i_k}, \vec{\theta}^{1}{}_{1,j_k}\big)\ , \dots , \sum_{k=1}^{K1} d\big(\vec{x}_{J,i_k}, \vec{\theta}^{1}{}_{J,j_k}\big)\Big), \dots ,$$
$$\frac{1}{K_G}\Big(\sum_{k=1}^{K_G} d\big(\vec{x}_{1,i_k}, \vec{\theta}^{G}{}_{1,j_k}\big)\ , \dots , \sum_{k=1}^{K_G} d\big(\vec{x}_{J,i_k}, \vec{\theta}^{G}{}_{J,j_k}\big)\Big)\Bigg]^T \qquad (5)$$

where $\vec{x}_{j,i_k}$ and $\vec{\theta}^{g}{}_{j,j_k}$ are 3D vectors specifying the positions of the j-th joint in the i_k-th frame of the action sample X and j_k-th frame of the reference sample of class g respectively, J is the number of joints, and K is the number of alignment index pairs. The reason we do not use the symbol $DTW(.)$ in the above equation is that the time series of each joint are not aligned separately between two action samples, but rather the same correspondence pairs derived previously, from the overall (multi-dimensional) alignment of the two action samples, are used.

B. Intra-Skeletal Features

The third category of features we introduce is the distance between the time series of joint pairs in an action sequence. There is no need for DTW here, and only the distances between the joints of a skeleton are averaged over time:

$$\Phi_3(\mathrm{X}) = \begin{bmatrix} avg\{d(\vec{x}_{1,i}, \vec{x}_{2,i})\}^{i:1..M}, \dots , avg\{d(\vec{x}_{1,i}, \vec{x}_{J,i})\}^{i:1..M}, \\ \dots , avg\{d(\vec{x}_{J-1,i}, \vec{x}_{J,i})^{i:1..M} \end{bmatrix}^T =$$

$$\frac{1}{M}\left[\sum_{i=1}^{M} d(\vec{x}_{1,i}, \vec{x}_{2,i})\ , \dots , \sum_{i=1}^{M} d(\vec{x}_{1,i}, \vec{x}_{J,i}), \dots , \sum_{i=1}^{M} d(\vec{x}_{J-1,i}, \vec{x}_{J,i})\right]^T \quad (6)$$

where M is the length of the action sample. Also, we add the standard deviation of the distances between the joints of a skeleton over time as $\Phi_4(\mathrm{X})$ to our features.

$$\Phi_4(X) = [std\{d(\vec{x}_{1,i}, \vec{x}_{2,i})\}^{i:1..M}, \dots , std\{d(\vec{x}_{1,i}, \vec{x}_{J,i})\}^{i:1..M}, \dots\}]^T \quad (7)$$

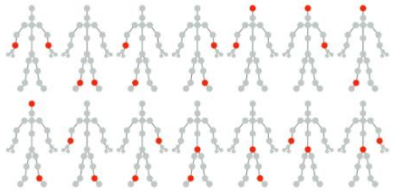
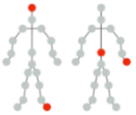
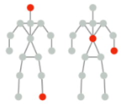
Since the distance between some joints, such as the elbow and the shoulder, does not change over time, here we consider only the pairs of joints whose distances vary, such as the right wrist and the right ankle, etc. These joints are shown in Table I, which will be discussed later.

C. Joint Importance Maps

For one of the four sets of features that we intend to extract from any given action, we propose using the idea of joint importance maps, to sort and select the most involved or discriminating joints in any action class.

For the joint pairs in (6) and (7), we use the same set of pairs for all actions and all datasets, as indicated in Table I. That is, we do not resort to a joint importance map, but instead use the symmetrical pairs of principal joints, as the intuitive choice.

TABLE I
JOINTS USED IN THE SECOND GROUP OF INTER-ACTION FEATURES AND THE INTRA-SKELETON FEATURES FOR THE FOUR DATASETS

Dataset	Joint pairs in the intra-skeleton features
TST	
UTKinect	Same as in TST dataset (with corresponding joints). For example, the first and last in the second row above become: 
UCFKinect	Same as in TST dataset (with corresponding joints). For example, the first and last in the second row above become: 
NTU RGB+D	Same as in TST dataset

Similarly, for features defined in (4), all joints are factored in, and there is again no intention for picking more important joints, as this feature is intended to sum the effects of "all" joints, both more- and less-important ones. The more detailed broken-down version of this feature will be that in (5), for which we do use joint importance maps.

The features defined in (5) could potentially include all individual joints, adding an extra entry to the features vector for any individual joint. However, as it can be readily seen, the size of the features vector increases rapidly with the number of joints incorporated (more specifically, as number of included joints × number of action classes.) We, therefore, intend to minimize the joints included in (5) to a minimum of joints having the most significance (as later defined.)

We proposed and experimented with two ideas of what would constitute an "important joint." One was based on computing cross-correlation matrices, between actions in each dataset, and all present joints. This yields a correlation matrix whose elements would suggest how strongly each joint is correlated with each action. This idea itself can be implemented in several ways, but we will not elaborate any more on this method as the second method demonstrates superior results.

The second method to arrive at important joints would be to simply tabulate the joint "movements" - more accurately, their variance about their mean positions - within each action class. In other words, in the second definition, we label joints with maximal movements (a corrected version of movements to be more specific) in each class, to be important joints for that class. This can be simply done by computing variances of individual joints throughout all frames, and average those over all given samples. Specifically, for example, for the UT-Kinect dataset, we have 20 samples (10 [subjects] x 2 [performances]) of each action, over which the across-the-frames variances will be averaged for each joint.

The reason this will give rise to an improvement in the recognition process is that we are in fact utilizing a priori knowledge about the actions: given our sample actions, we could know that not all joints engage equally or are equally important in a specific action; those with a higher (corrected version of) variance contribute more to the "information" in each action.

The reason we mention "corrected version" of variance is that, it will be unfair, so to say, to compare raw variances of joints, as for example with long limbs the variances are already high at distal points compared to proximal points, given the swing distance they get, even if they undergo slight rotations. Said otherwise, and talking about a certain limb like the arm, the elbow might be more important than the hand end-point, even if it always sustains less variance compared to the hand, simply because the hand has a longer swing radius from the shoulder.

It seems that there are 3 principal ways in which a joint could get such swing "advantage" – two from radii of rotation about the horizontal and vertical axes, and one from free limb swing (referred to above.) For example, talking about the elbow, there's a component due to possible rotation about the vertical centerline (passing through the hip, parallel to the spine) due to non-zero distance (radius) from the vertical centerline, as in swinging the arms around the body with shoulders kept still, and a second component of free-swing motion, due to swing distance from the shoulder, as in performing jumping jacks. The larger the radius from the vertical axis in the first case, the larger the swing due to any spin about the vertical centerline; and the farther from the shoulder on the arm in the second case, the larger the free swing radius. The free swing component is only associated to freely swinging limbs: the arms, legs, and the head.

As another example, for the case of the head joint, there's zero component of first swing type (due to zero distance to vertical axis, with head lying on the axis), but non-zero components of the other two: the head is connected to the neck joint, at the highest radius, and undergoes maximal movement as the neck moves, more so than say the eyes or chin, ergo non-zero free limb swing component, and another component regarding rotation about the horizontal axis (passing through the hips), as when the person bends forward with legs kept straight. More generally, for the final component, the hip gets a radius of zero, and any joint more vertically distant gets greater values, the neck getting higher values than the chest for example.

We can therefore attribute an "equivalent radius" to any joint, describing the swing advantage it gets. We used the average skeleton in the standing posture (the first frames in our action samples), to calculate the three radii, and thereupon the equivalent radii. Quantitatively, for example in the TST dataset, the head gets a radius of 0 cm on the first component (as explained above), 28 cm on the swing component (distance from the head to the neck joint), and 83 cm on the bending radius (distance from the head to the hip joint), or compactly (0, 28, 83) cm; the elbow gets (23, 30, 24) cm correspondingly. We sum the three radii and report the square root as the correction factor in Table II: $\sqrt{0+28+83} \cong 10.53$ and $\sqrt{23+30+24} \cong 8.77$, for joints 4 (head) and 6 (elbow) respectively. Note that the three radii are not independent (or orthogonal), and therefore superiority of norm-2 compared to summation is not straightforward.

It should be mentioned that this is a gross approximation of the dynamics involved and possibly not the best way to account for these rotations, and they might be more rigorously addressed using mechanical engineering concepts. It should be noted that the authors had an intuitive idea what the final overall swing factors for each joint should look like, and adopted formulae to maximally approach that.

TABLE II

EXAMPLE CALCULATION OF CORRECTION FACTORS FOR JOINTS: WHAT RAW VARIANCES NEED TO BE DIVIDED BY FOR FAIR COMPARISON OF JOINTS

TST				
1: --*	2: 5.91	3: 8.24	4: 10.53	5: 8.544
6: 8.77	7: 9.16	8: 9.84	9: 8.24	10: 8.60
11: 9.16	12: 9.84	13: 2.82	14: 10.39	15: 13.67
16: 14.07	17: 2.82	18: 10.77	19: 13.78	20: 14.35
21: 7.92				
UTKinect				
1: --*	2: 2.92	3: 7.04	4: 9.49	5: 7.01
6: 7.73	7: 8.85	8: 9.47	9: 6.99	10: 7.56
11: 9.28	12: 10.00	13: 3.66	14: 10.37	15: 13.19
16: 13.51	17: 3.67	18: 9.45	19: 12.04	20: 12.50
UCF				
1: 9.39	2: 6.64	3: --*	4: 7.73	5: 8.38
6: 10.55	7: 7.72	8: 8.23	9: 10.17	10: 3.26
11: 9.70	12: 13.19	13: 3.17	14: 9.58	15: 12.99
NTU-RGB+D				
1: --*	2: 5.91	3: 8.10	4: 10.12	5: 8.62
6: 8.85	7: 9.26	8: 9.73	9: 8.45	10: 8.62
11: 9.30	12: 9.80	13: 3.06	14: 9.23	15: 13.48
16: 14.27	17: 2.88	18: 10.53	19: 14.02	20: 14.41
21: 7.23				

* No division needed. The coordinates are zero.

The joint movements (variances) should now be divided by these factors first, to normalize for and cancel out the effect of swing advantages, before sorting to select most important joints. In other words, we get a more "fair" comparison of joint variances, if we first divide raw variances by these factors. The procedure to arrive at final joint importance maps is thereupon straightforward; We compute the average joint variances over the entire dataset for each action class and put them in a matrix. We correct these variances by dividing them by their corresponding correction factors in Table II. We now sort the corrected variances and use the first most important ones for computing features in equation (5).

Equivalently, the factors suggest how much each joint is inherently susceptible to movement, and very well conform to intuition. For example, values 8.77 and 9.84 corresponding to elbow (Joint 6) and hand (Joint 8) in TST table, suggest that the hand inherently gets more swing, as when the whole arm is moving in a uniform circular manner, partially due to the fact that the radius to the hand is larger than that to the elbow.

The tables should ideally be symmetrical, with factors in, say, left and right knee being equal; any deviation is stemming from the fact that we've estimated these from our action samples. Furthermore, such factors should be independent of datasets (when joint definitions are the same), as they solely depend on the average human anatomy and position of joints. Given any new dataset, with a new set of joint definitions, one could refer to a simple human anatomy model, and make manual measurements, to construct the tables. We have used one or average of several frames (skeletons) from the datasets in the standing posture, to rid us of the need to make manual measurements, in constructing Table II. It should be evident that furthermore, the absolute value of these factors doesn't matter but their ratios, as variances will be 'divided' by these before sorting. The numbers for the NTU dataset have thus been scaled to get equal values for joint 2 with TST.

For any new dataset, the above approach adds a single extra step to the processing pipeline: one or more samples in the standing posture are needed to estimate relative joint positions, to be followed by computation of the three radii based on joint coordinates (using only x coordinates for one radius, only y coordinates for the other radius, and distances from the shoulders, hips, and the neck joint for the free swing radius), forming final equivalent radii, and using them as correction factors for variance adjustment.

Table III reports the most important joints thus found. Upon adjustment by the factors, and sorting the joints, the top most important joints are selected. The 'number' of most important joints used is different in each dataset, determined empirically to yield the best results. This number has to do with other parameters in the dataset, including complexity and similarity of the actions, the level of noise, and presence of joints spurious movements. The UTKinect dataset, being a relatively easy dataset for classification, can do well with 3 joints only for example. The joints very well correspond to what are intuitively considered to be the most engaged joints in an action.

D. Classification Using DTW-Based Features and Selection of Reference Sample

The four groups of features described above are concatenated to form the final feature vector as

$$\Phi(\mathrm{X}) = [\alpha_1\Phi_1(\mathrm{X})^T, \alpha_2\Phi_2(\mathrm{X})^T, \alpha_3\Phi_3(\mathrm{X})^T, \alpha_4\Phi_4(\mathrm{X})^T]^T \qquad (8)$$

in which α_i are coefficients the categories might need due to their different scales. These coefficients will be adjusted through k-fold cross-validation on the training set. Finally, the classification of actions can be done using a suitable classifier. Here we use the regularized Fisher LDA method because of its high classification power while being computationally efficient. Given that the number of samples is usually not significant in relation to the number of features, there is a risk of overfitting, and using the shrinkage parameter dramatically reduces this risk, and leads to a classifier with relatively stable parameters. It should be noted that elaborating on the classifier itself is not the focus of this article, and rather the main purpose has been extracting proper features that are robust to within-class changes.

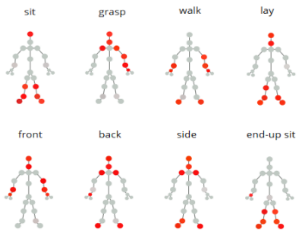
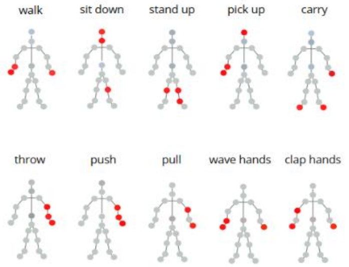
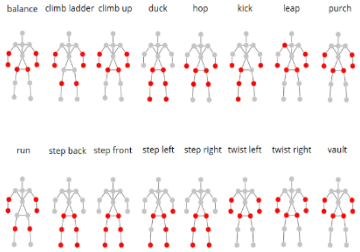
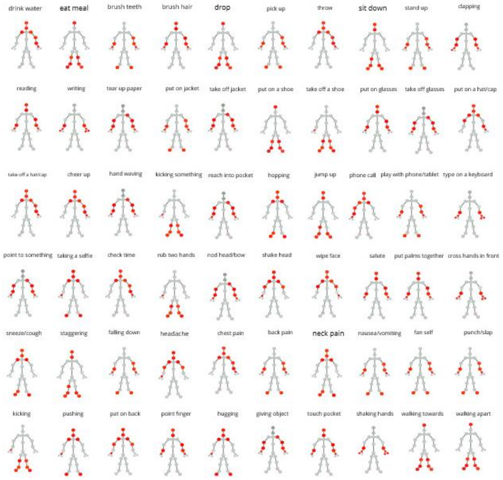
As stated in Section III, in the proposed method, one sample among the training samples in each class is selected as a reference. Then the DTW-based features are extracted concerning the reference sample. In this paper, various methods for selecting the optimal reference sample have been investigated and evaluated.

One of the methods tested was to search for the sample that was the mean of the samples of a class, in terms of DTW distance (in this definition, the sample that has the least sum of squares of distances from all other samples of a class, is considered to be the mean sample of that class). Another method was to select the best sample in terms of a Fisher-like criterion; that is, choosing the sample that has the maximum ratio of the sum of squares of distances from the samples of other classes to the sum of squares of distances from the samples of its own class. In another approach, the optimal reference sample was searched for a thorough evaluation of each training set. However, ultimately, the method that yielded the best result was a random selection of the reference sample from the samples of a class. Since this method further imparts a stochastic property to the classifier, it can be used to combine several classifiers, each of which has different reference samples to improve the final accuracy. We use score level fusion to combine different classifiers, by simply adding together the scores of different classifiers for each class.

E. Computational complexity:

The computational cost of this algorithm mainly lies in the need to calculate DTW distances between the query sample and reference sequences for each class of training samples. So, it can be succinctly approximated by $O(G \times N \times p_1 \times p_2)$ where G is the number of classes, N is the number of joints per skeleton, p_1 denotes the length of the query sample and p_2 is the minimum length of the training references. For comparison, we can take a look at some classical and deep learning based strategies introduced in recent years. As for Switonski's algorithm [25], it directly uses the vanilla DTW, that has a complexity of $O(M \times N \times p_1 \times p_2)$, where M is number of training samples. Therefore, it suggests that our DTW based method is considered to be a less time consuming algorithm than those that generally fall into the vanilla DTW based approaches. For Slama's [14], one can observe that the time complexity is at least as high as the cost of the vanilla DTW even ignoring all those incurred by calculating the ARMA modeling and representations of linear subspaces. For a row skeleton based LSTM network, the major cost can be approximated by $O(p \times N_c \times (N_c + N_i + N_o))$, where p is the maximum length of sequences, N_c is the number of memory cells, N_i is the number of input units, and N_o is the number of output units.

TABLE III
LIST OF CALCULATED MOST IMPORTANT JOINTS, BROKEN DOWN BY ACTION CLASS, FOR ALL DATASETS

Dataset	
TST	Seven most important joints:  sit grasp walk lay front back side end-up sit
UTKinect	Three most important joints:  walk sit down stand up pick up carry throw push pull wave hands clap hands
UCFKinect	Six most important joints:  balance climb ladder climb up duck hop kick leap purch run step back step front step left step right twist left twist right vault
NTU RGB+D	Seven most important joints:  drink water eat meal brush teeth brush hair drop pick up throw sit down stand up clapping reading writing tear up paper put on jacket take off jacket put on a shoe take off a shoe put on glasses take off glasses put on a hat/cap take off a hat/cap cheer up hand waving kicking something reach into pocket hopping jump up phone call play with phone/tablet type on a keyboard point to something taking a selfie check time rub two hands nod head/bow shake head wipe face salute put palms together cross hands in front sneeze/cough staggering falling down headache chest pain back pain neck pain nausea/vomiting fan self punch/slap kicking pushing pat on back point finger hugging giving object touch pocket shaking hands walking towards walking apart

V. EXPERIMENTS

In this section, we will evaluate the proposed method on three well-known action datasets and compare the results with the state-of-the-art. The selected datasets are TST-Fall, UTKinect, and UCFKinect [11, 31, 32]. These datasets include a tasty variety of different action types. The TST-Fall dataset includes involuntary actions of falling, which naturally involve considerable interpersonal differences. The UTKinect dataset includes daily actions that have a moderate level of interpersonal differences. The UCFKinect dataset includes show actions that are suitable for games or human-computer interactions. Each of these datasets is described in more detail below. It is important to note that existing methods often focus on one dataset and fine-tune their method on that specific dataset. However, a method that can function considerably well on different datasets is more generalizable and more applicable.

The evaluation method in this research on all datasets is leave-one-subject-out, in which the algorithm is repeated by the number of people in the dataset, each time the actions of one person is left out and the actions of others are considered as training, and finally, the recognition rates are averaged over all trials.

In each run, the parameters in the method, which estimate with grid search cross validation, including the shrinkage factor and the coefficients of the feature groups, are obtained through evaluation on the training set.

To account for the detrimental effect of direction of motion or angle relative to the sensor, skeletons need to be aligned. Two transformations are made to align skeletons in all frames. One is translating the hip joint to the origin. The other is rotating the skeleton about the y axis so that the projection of the line connecting the shoulder joints in the xz plane is parallel to the x-axis. These are in Figure 2.

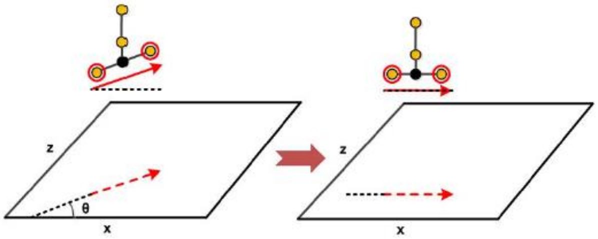


Fig. 2. Alignment of skeletons

A. TST Dataset

The TST Fall dataset [31] includes two categories of activities of daily living and involuntary fall actions. The daily living category includes sit, grasp an object, walk, and lie down, and the fall category includes various fall actions, including fall forward, back, on the side, and ends-up sitting fall. Each action has been performed three times by 11 individuals, and as a result, this dataset contains 264 action samples. In addition to the depth data recorded by the Kinect sensor, wearable sensor data is also available in this dataset, which provides valuable information. However, given the purpose of this article, only depth information has been used here for action recognition.

Joints available in the skeletal data in this dataset are shown in Figure 3. Joints 22-25 have not been used in the proposed method since they are not engaged in the actions in this dataset. Also, the joints used to align the keleton in successive action frames are specified in this figure.

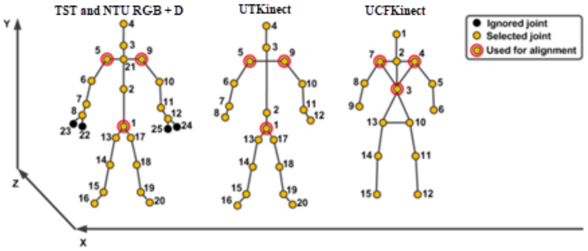


Fig. 3. Joints in skeletal data for each of the datasets along with coordinate axes.

The involuntary actions in this dataset have made it one of the most challenging datasets, and methods that perform successfully on this dataset will be beneficial, especially for applications in caring for the sick and the elderly.

B. UTKinect Dataset

The UTKinect dataset [11] includes ten action types: walk, sit down, stand up, pick up, carry, throw, push, pull, wave hands, clap hands. Each action is performed twice by ten people. Since one of the samples of the action 'carry' in fact contains a different action, we have removed this sample from the data. This omission is also done in other articles that have used this dataset for evaluation. Therefore, the UTKinect dataset used includes 199 action samples. Joints in skeletal data in this dataset are shown in Figure 3.

The relatively high variance in performing each action by different individuals is one of the challenges of this dataset. Another issue with this dataset is the noise in joint position measurements, causing unwanted jitters in the joints positions. This noise is another challenge in this dataset.

C. UCFKinect Dataset

The UCFKinect dataset [32] includes 16 relatively small actions, and each action performed five times by 16 individuals. There are a total of 1280 action samples in this dataset. Joints in skeletal data in this dataset are shown in Figure 3.

This dataset is mainly compiled for simulating actions used in games. The actions in this dataset are balanced, climb a ladder, climb up, duck, hop, kick, leap, punch, run, step back, step front, step left, step right, twist left, twist right, and vault.

D. NTU RGB+D Dataset

This database [33] has been created in response to increasing demands for a large scale dataset suitable to be used for data hungry algorithms like those based on deep learning. The database includes 56880 sequences from 40 different subjects acquired in different settings of the camera. Similar to TST dataset, the skeletal information consists of the 3D location of 25 major joints over time. There are 60 categories of actions including 40 daily activities, 11 interactions and 9 medical conditions. All samples have been collected using a Kinect v2 sensor under different environments.

E. Visualizing the Action Sequences in the Skeleton State Space

We present a method to visualize the action sequences. This approach has several other benefits as well, including helping to visualize the sequence of action in one glance, without having to examine individual frames. We do the plot in the two-dimensional plane since it is more suitable for visualization. To do this, first, we place the joint positions in each frame in a vector. For example, if the skeleton has 20 joints since the position of each joint has three dimensions, we will have a 60-dimensional vector for each frame. We call these vectors the skeletal vectors. We use the skeletal vectors of all frames of action samples of one individual in the dataset to train a PCA. The resultant PCA space, although trained by samples of only one individual, can very well describe the skeletal vectors of actions of other individuals. We use the first two components of PCA to visualize skeletal vectors in all frames for each action sample, and thus plot the action sequence in a two-dimensional space. We call this space the skeleton state space. Figure 4 bottom shows the sequences corresponding to three samples of the action' lie down' performed by individual #3, and on the top, three samples of the action 'ends-up sitting fall' performed by individual #11 in the TST dataset.

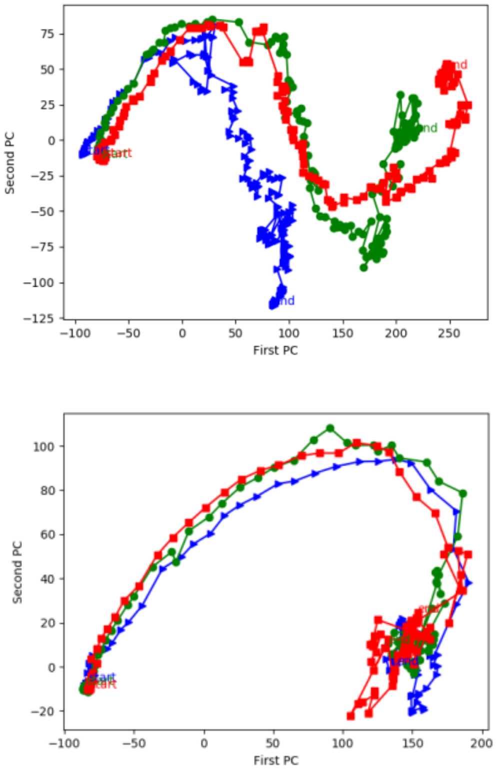


Fig. 4. Action sequences corresponding to 'lie down' (down) and 'ends-up-sitting fall' (up) in the TST dataset performed by individuals #3 and #11, respectively. Each action has been performed three times. The start and endpoints of sequences have been labeled so.

One of the critical benefits of this visualization is finding outlier samples in a dataset. For example, in Figure 5, the sequences of three samples of the action' lie down' performed by individual #1 in the TST dataset are plotted. Upon plotting these sequences, we found that since in the plot, the start and endpoints of the sequences lie very close to each other, this person, unlike the ten others in the dataset (refer to Figure 4 bottom), returns to his original position after performing the action. The validity of this conjecture was later confirmed by checking the original sequences of frames of this individual. However, we did not remove these samples from the dataset to test the method's robustness to outlier samples.

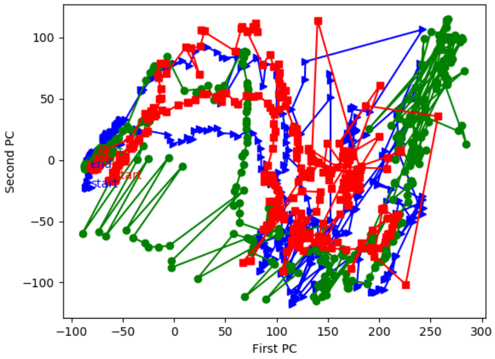


Fig. 5. Sequences of three performances of the action ‘lie down’ in the TST dataset performed by individual #1. The person returns to his original position after lying down.

Visualizing the sequences can also well illustrate the capability of the DTW method in creating a correspondence between frames in two sequences. Figure 6 top shows the correspondences before DTW between two sequences of action 'lie down' in the TST dataset performed by individuals #2 and #3. In this figure, correspondences between frames in the two sequences in the skeleton state space are shown in blue lines. The correspondences after DTW between the two sequences are shown in Figure 6 bottom, which demonstrate the superiority of DTW in establishing correspondence.

F. Evaluation

Since the numbering of the joints is not the same in the three datasets, Table I lists the joints used in the second group of inter-action features and Table III lists those in the intra-skeleton features for each of the three datasets.

The recognition rate of the proposed method on the TST dataset, along with the recognition rates of the state-of-the-art are shown in Table IV. The confusion matrix of action recognition on this dataset using the proposed method is shown in Figure 7A.

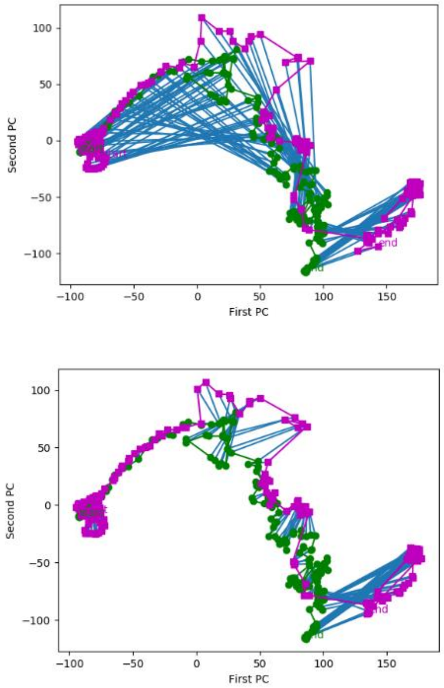


Fig. 6. Correspondence between frames in two sequences of ‘lie down’ performed by individuals #2 and #3 in the TST dataset. Correspondences using frame indices (up), and correspondences are resulting from DTW (down), both shown in blue lines.

TABLE IV
Recognition Rate of the Proposed Method Compared to Other Methods on the TST Dataset

Method	Recognition Rate	Year
Ghojogh et al. [12]	88.6%	2018
Ghodsi et al. [34]	92.3%	2018
Seredin [31]	91.7%	2019
Our method	94.55%±0.04%	-

It should be noted that some of the rates reported in the literature for this dataset have been obtained using wearable sensor data and therefore are not listed in this table due to irrelevance.

The recognition rate of the proposed method and the state-of-the-art on the UTKinect dataset are shown in Table V. It should be noted that, as shown in Table III, only the head, neck, and spine joints are used in the second group of inter-action features in this dataset. Using more joints in this group slightly reduces the recognition rate. The reason could be the noise in the joint position estimations in the dataset that become more pronounced when the joints are used separately. The confusion matrix of action recognition using the proposed method on this dataset is shown in Figure 7B.

TABLE V
Recognition rate of the proposed method compared to other methods on the UTKinect dataset

Method	Recognition Rate	Year
Vemulapalli et al. [13]	97.0%	2014
Antunes et al. [35]	95.1%	2016
Gupta and Bhavsar [36]	96.0%	2016
Ghodsi et al. [34]	96.8%	2018
Rhif et AL. [37]	96.68%	2018
Xiang Gao [38]]	98.5%	2019
Our method	98.9%±0.3%	-

It is worth noting that rates such as 98.5% [39], 98.8% [40], and 99.19% [41] have also been reported in other papers for this dataset, which has not been included in the table. The reason for this exclusion is that in their experiments in these papers, the actions of all individuals in the dataset are mixed, and then some percentage of them are randomly selected for testing. As a consequence, samples of the actions of test individuals are included in the training set, which unfairly helps the classifier in learning.

This is while the main challenge in action recognition recognizes the actions of individuals unseen by the system.

The recognition rate of the proposed method and the state-of-the-art on the UCFKinect dataset are shown in Table VI.

Again, rates such as 97.9% [14] and 98.7% [42] for this dataset, which has been reported by dividing 'samples' for training and test rather than dividing 'individuals', are not included in the table. The confusion matrix of action recognition using the proposed method on this dataset is shown in Figure 7C.

TABLE VI
Recognition rate of the proposed method compared to other methods on the UCFKinect dataset

Method	Recognition Rate	Year
Zanfir et al. [43]	98.5%	2013
Kerola et al. [44]	98.8%	2014
Yang et al. [17]	97.1%	2014
Beh et al. [45]	98.9%	2014
Ding et al. [46]	98.0%	2015
Lu et al. [18]	97.6%	2016
Ghodsi et al. [34]	97.9%	2018
sun bin [47]	98.91%	2019
Our method	99.14%±0.01%	-

Finally, the recognition rate of the proposed method on the NTU RGB+D dataset, along with the recognition rates of the state-of-the-art are shown in Table VII. The confusion matrix of action recognition using the proposed method on this dataset is shown in Figure 8.

TABLE VII
Recognition rate of the proposed method compared to other methods on the NTU RGB+D dataset

Method	Recognition Rate	Year
Zhang et al. [48]	95.0%	2018
Shi et al. [49]	95.1%	2019
Shi et al. [50]	96.1%	2019
Our method	97.85%±0.14%	-

In comparison to methods based on deep learning, our method is doing pretty well. The LSTM based method in [37] reports an accuracy of 96.68% in cross validation on the UTKinect dataset while we report an accuracy of 98.9%. The. method in [51] reports a best accuracy of 96.1% on NTU RGB+D, while we achieve 97.8%. Again methods, such as [52], using wearable sensor data are not reported.

G. The Effect of Selection of Important Joints

To demonstrate the effect of the important joints scheme, as compared to selection of 'all' joints, we have constructed Table VIII. The first column corresponds to our proposed method using the important joints, while the number of important joints used in the second column has been set to the total number of joints, which is equivalent to using all joints. It can be seen that the performance deteriorates, consistent with intuition; the performance should quickly deteriorate as more joints are included as the number of features grows very quickly with the addition of each new joint (more specifically, at least by the number of classes times the number of joints), which hasten a curse of dimensionality, as the number of our training samples cannot grow correspondingly.

H. The Effect of the Number of Base Classifiers in the Ensemble

In Figure 9, the mean and standard deviation of the classification accuracy for the four datasets versus the number of combined classifiers are plotted. As can be seen, the TST dataset poses the most significant challenge due to the involuntary actions that introduce a high variance within action classes. Another point to notice is that in the UCFKinect dataset, due to the abundance of training samples, even one classifier yields excellent results.

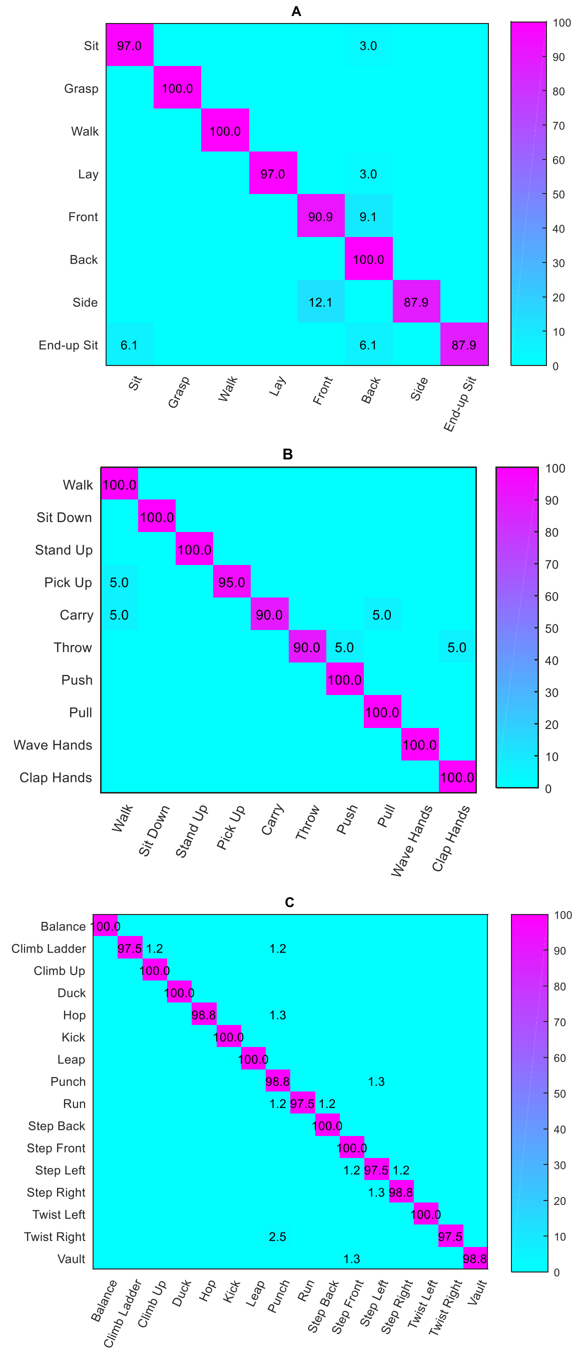


Fig. 7. Confusion matrices of action recognition for (A) TST, (B) UTKinect, and (C) UCFKinect dataset

TABLE VIII
INVESTIGATING THE SIGNIFICANCE OF SELECTING "IMPORTANT JOINTS" COMPARED TO "ALL JOINTS"

Dataset ↓	Recognition Rate of Our Method	
	Using Important Joints	Using All Joints
UTKinect	98.9%±0.3%	96.1%±0.01%
UCF	99.14%±0.01%	98.08%±0.03%
TST	94.55%±0.04%	92.33%±0.02%
NTU RGB+D	97.85%±0.14%	94.65%±0.04%

I. Feature Importance

To evaluate the importance of each group of proposed features and their various combinations, we calculated the classification accuracy in each case. The results are shown in Figure 10. At it can be seen, Φ_1 and Φ_2 are more important than the other two groups. Also, when either of Φ_1 or Φ_2 are used with another group of feature, a reasonably high accuracy is obtained. Finally, using all four groups, results in the best accuracy.

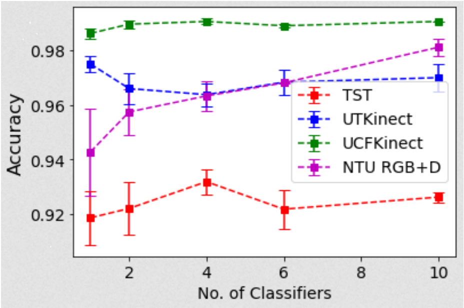


Fig. 9. Accuracy of the final classifier with respect to the number of combined classifiers

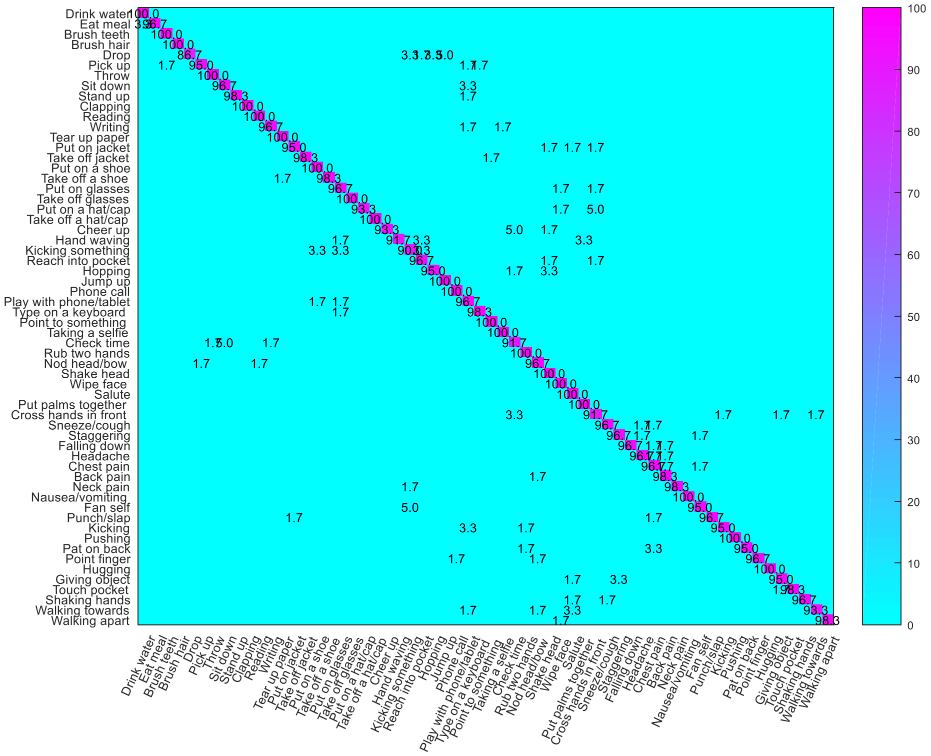


Fig. 8. Confusion matrix of action recognition for NTU RGB+D dataset

.

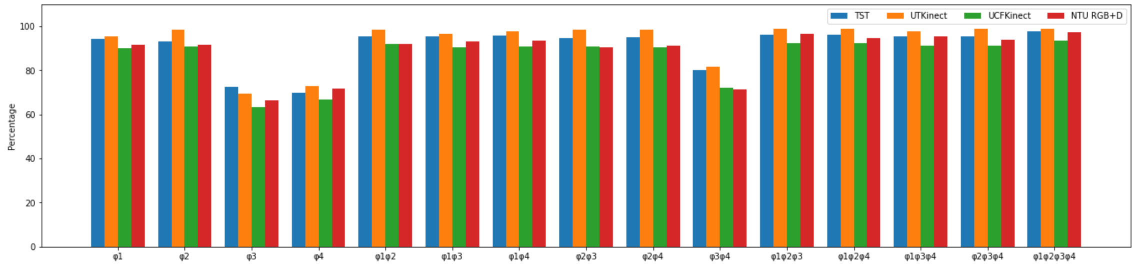


Fig. 10. Classification accuracy for various combinations of the four groups of proposed features using TST, UTKinect, UCF , and NTU-RGB+D dataset

VI. Conclusion

In this paper, an action recognition method using skeletal data extracted from depth images is presented. The proposed method treats the motion sequences as multi-dimensional time series and uses the DTW metric to map the series to a vector space. This method can simultaneously address the problem of nonlinear distribution of the motion sequences, and the problem of their temporal variations using the created vector space. Since the proposed features are calculated only with respect to the reference samples of each class, its computational complexity is drastically reduced compared with other methods based on DTW. The evaluations on the four datasets TST, UTKinect, UCFKinect, and NTU-RGB+D show the competitive performance of the proposed method with the state-of-the-art methods in this area.

References

[1] H.-B. Zhang *et al.*, "A comprehensive survey of vision-based human action recognition methods," *Sensors,* vol. 19, no. 5, p. 1005, 2019.

[2] S. Z. Gurbuz and M. G. Amin, "Radar-based human-motion recognition with deep learning: Promising applications for indoor monitoring," *IEEE Signal Processing Magazine,* vol. 36, no. 4, pp. 16-28, 2019.

[3] Z. Hussain, M. Sheng, and W. E. Zhang, "Different approaches for human activity recognition: A survey," *arXiv preprint arXiv:1906.05074,* 2019.

[4] M. Piriyajitakonkij *et al.*, "SleepPoseNet: Multi-view learning for sleep postural transition recognition using UWB," *IEEE Journal of Biomedical and Health Informatics,* 2020.

[5] J. Shotton *et al.*, "Real-time human pose recognition in parts from single depth images," *Communications of the ACM,* vol. 56, no. 1, pp. 116-124, 2013.

[6] S. Riofrío, D. Pozo, J. Rosero, and J. Vásquez, "Gesture recognition using dynamic time warping and kinect: A practical approach," in *2017 International Conference on Information Systems and Computer Science (INCISCOS)*, 2017: IEEE, pp. 302-308.

[7] A. D. Calin, "Gesture recognition on kinect time series data using dynamic time warping and hidden markov models," in *2016 18th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, 2016: IEEE, pp. 264-271.

[8] S. Sempena, N. U. Maulidevi, and P. R. Aryan, "Human action recognition using dynamic time warping," in *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics*, 2011: IEEE, pp. 1-5.

[9] F. Lv and R. Nevatia, "Recognition and segmentation of 3-d human action using hmm and multi-class adaboost," in *European conference on computer vision*, 2006: Springer, pp. 359-372.

[10] X. Li, Y. Zhang, and D. Liao, "Mining key skeleton poses with latent svm for action recognition," *Applied Computational Intelligence and Soft Computing,* vol. 2017, 2017.

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

[12] B. Ghojogh, H. Mohammadzade, and M. Mokari, "Fisherposes for human action recognition using Kinect sensor data," *IEEE Sensors Journal,* vol. 18, no. 4, pp. 1612-1627, 2017.

[13] R. Vemulapalli, F. Arrate, and R. Chellappa, "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*, 2014, pp. 588-595.

[14] R. Slama, H. Wannous, M. Daoudi, and A. Srivastava, "Accurate 3D action recognition using learning on the Grassmann manifold," *Pattern Recognition,* vol. 48, no. 2, pp. 556-567, 2015.

[15] Y. Du, W. Wang, and L. Wang, "Hierarchical recurrent neural network for skeleton based action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1110-1118.

[16] G. Evangelidis, G. Singh, and R. Horaud, "Skeletal quads: Human action recognition using joint quadruples," in *2014 22nd International Conference on Pattern Recognition*, 2014: IEEE, pp. 4513-4518.

[17] X. Yang and Y. Tian, "Effective 3d action recognition using eigenjoints," *Journal of Visual Communication and Image Representation,* vol. 25, no. 1, pp. 2-11, 2014.

[18] G. Lu, Y. Zhou, X. Li, and M. Kudo, "Efficient action recognition via local position offset of 3D skeletal body joints," *Multimedia Tools and Applications,* vol. 75, no. 6, pp. 3479-3494, 2016.

[19] M. A. Gowayyed, M. Torki, M. E. Hussein, and M. El-Saban, "Histogram of oriented displacements (HOD): Describing trajectories of human joints for action recognition," in *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.

[20] Y. Guo, Y. Li, and Z. Shao, "DSRF: A flexible trajectory descriptor for articulated human action recognition," *Pattern Recognition,* vol. 76, pp. 137-148, 2018.

[21] C. Wang, Y. Wang, and A. L. Yuille, "An approach to pose-based action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 915-922.

[22] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy, "Sequence of the most informative joints (smij): A new representation for human skeletal action recognition," *Journal of Visual Communication and Image Representation,* vol. 25, no. 1, pp. 24-38, 2014.

[23] M. Jiang, J. Kong, G. Bebis, and H. Huo, "Informative joints based human action recognition using skeleton contexts," *Signal Processing: Image Communication,* vol. 33, pp. 29-40, 2015.

[24] J. Weng, C. Weng, J. Yuan, and Z. Liu, "Discriminative spatio-temporal pattern discovery for 3D action recognition," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 29, no. 4, pp. 1077-1089, 2018.

[25] A. Switonski, H. Josinski, and K. Wojciechowski, "Dynamic time warping in classification and selection of motion capture data," *Multidimensional Systems and Signal Processing,* vol. 30, no. 3, pp. 1437-1468, 2019.

[26] H.-R. Choi and T. Kim, "Modified dynamic time warping based on direction similarity for fast gesture recognition," *Mathematical Problems in Engineering,* vol. 2018, 2018.

[27] J. Rwigema, H.-R. Choi, and T. Kim, "A differential evolution approach to optimize weights of dynamic time warping for multi-sensor based gesture recognition," *Sensors,* vol. 19, no. 5, p. 1007, 2019.

[28] H. Mohammadzade and M. Tabejamaat, "Sparseness embedding in bending of space and time; a case study on unsupervised 3D action recognition," *Journal of Visual Communication and Image Representation,* vol. 66, p. 102691, 2020.

[29] P. Bojanowski *et al.*, "Weakly supervised action labeling in videos under ordering constraints," in *European Conference on Computer Vision*, 2014: Springer, pp. 628-643.

[30] E. Keogh and C. A. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowledge and information systems,* vol. 7, no. 3, pp. 358-386, 2005.

[31] O. Seredin, A. Kopylov, S.-C. Huang, and D. Rodionov, "A SKELETON FEATURES-BASED FALL DETECTION USING MICROSOFT KINECT V2 WITH ONE CLASS-CLASSIFIER OUTLIER REMOVAL," *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences,* 2019.

[32] C. Ellis, S. Z. Masood, M. F. Tappen, J. J. LaViola, and R. Sukthankar, "Exploring the trade-off between accuracy and observational latency in action recognition," *International Journal of Computer Vision,* vol. 101, no. 3, pp. 420-436, 2013.

[33] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "Ntu rgb+ d: A large scale dataset for 3d human activity analysis," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1010-1019.

[34] S. Ghodsi, H. Mohammadzade, and E. Korki, "Simultaneous joint and object trajectory templates for human activity recognition from 3-D data," *Journal of Visual Communication and Image Representation*, vol. 55, pp. 729-741, 2018.

[35] M. Antunes, D. Aouada, and B. Ottersten, "A revisit to human action recognition from depth sequences: Guided

svm-sampling for joint selection," in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2016: IEEE, pp. 1-8.

[36] K. Gupta and A. Bhavsar, "Scale invariant human action detection from depth cameras using class templates," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016, pp. 38-45.

[37] M. Rhif, H. Wannous, and I. R. Farah, "Action recognition from 3D skeleton sequences using deep networks on lie group features," in *2018 24th International Conference on Pattern Recognition (ICPR)*, 2018: IEEE, pp. 3427-3432.

[38] X. Gao, W. Hu, J. Tang, J. Liu, and Z. Guo, "Optimized skeleton-based action recognition via sparsified graph regression," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 601-610.

[39] J. Liu, G. Wang, P. Hu, L.-Y. Duan, and A. C. Kot, "Global context-aware attention lstm networks for 3d action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1647-1656.

[40] Y. Yang, C. Deng, D. Tao, S. Zhang, W. Liu, and X. Gao, "Latent max-margin multitask learning with skelets for 3-D action recognition," *IEEE transactions on cybernetics*, vol. 47, no. 2, pp. 439-448, 2016.

[41] Y. Yang, C. Deng, S. Gao, W. Liu, D. Tao, and X. Gao, "Discriminative multi-instance multitask learning for 3d action recognition," *IEEE Transactions on Multimedia*, vol. 19, no. 3, pp. 519-529, 2016.

[42] X. Jiang, F. Zhong, Q. Peng, and X. Qin, "Online robust action recognition based on a hierarchical model," *The Visual Computer*, vol. 30, no. 9, pp. 1021-1033, 2014.

[43] M. Zanfir, M. Leordeanu, and C. Sminchisescu, "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*, 2013, pp. 2752-2759.

[44] T. Kerola, N. Inoue, and K. Shinoda, "Spectral graph skeletons for 3D action recognition," in *Asian conference on computer vision*, 2014: Springer, pp. 417-432.

[45] J. Beh, D. K. Han, R. Durasiwami, and H. Ko, "Hidden Markov model on a unit hypersphere space for gesture trajectory recognition," *Pattern Recognition Letters*, vol. 36, pp. 144-153, 2014.

[46] W. Ding, K. Liu, F. Cheng, and J. Zhang, "STFC: Spatio-temporal feature chain for skeleton-based human action recognition," *Journal of Visual Communication and Image Representation*, vol. 26, pp. 329-337, 2015.

[47] B. Sun, D. Kong, S. Wang, L. Wang, Y. Wang, and B. Yin, "Effective human action recognition using global and local offsets of skeleton joints," *Multimedia Tools and Applications*, vol. 78, no. 5, pp. 6329-6353, 2019.

[48] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, and N. Zheng, "View adaptive neural networks for high performance skeleton-based human action recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 8, pp. 1963-1978, 2019.

[49] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12026-12035.

[50] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Skeleton-based action recognition with directed graph neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7912-7921.

[51] B. Ren, M. Liu, R. Ding, and H. Liu, "A Survey on 3D Skeleton-Based Action Recognition Using Learning Method," *arXiv preprint arXiv:2002.05907*, 2020.

[52] T.-H. Tsai and C.-W. Hsu, "Implementation of Fall Detection System Based on 3D Skeleton for Deep Learning Technique," *IEEE Access*, vol. 7, pp. 153049-153059, 2019.