J. Vis. Commun. Image R. 30 (2015) 10–21

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Joint movement similarities for robust 3D action recognition using skeletal data ☆

Hossein Pazhoumand-Dar *, Chiou-Peng Lam, Martin Masek

School of Computer and Security Science, Edith Cowan University, Perth, WA 6050, Australia

ARTICLE INFO

Article history:
Received 17 September 2014
Accepted 5 March 2015
Available online 16 March 2015

Keywords:
Human action recognition
Similarity function
Longest common subsequence algorithm
Kinect camera
Motion capture system
Discriminative features
Motion pattern
Trajectory modeling

ABSTRACT

Human action analysis based on 3D imaging is an emerging topic. This paper presents an approach for the problem of action recognition using information from a number of action descriptors calculated from a skeleton fitted to the body of a tracked subject. In the proposed approach, a novel technique that automatically determines discriminative sequences of relative joint positions for each action class is employed. In addition, we use an extended formulation of the longest common subsequence algorithm as a similarity function, which allows the classifier to reliably find the best match for extracted features from noisy skeletal data. The proposed approach is evaluated using two existing datasets from the literature, one captured using a Microsoft Kinect camera and the other using a motion capture system. The experimental results show that the approach outperforms existing skeleton-based algorithms in terms of its classification accuracy and is more robust in the presence of noise when compared to the dynamic time warping algorithm for human action recognition.

© 2015 Elsevier Inc. All rights reserved.

1. Introduction

The task of human action recognition has been studied over the last few decades as an important area of computer vision research. Its applications include surveillance [1], robot control [2], human computer interaction [3], and healthcare [4]. Generally, human motions during known actions are represented via extracted features, and then recognition of a new instance is performed by comparing and classifying them using these representations.

With the advent of inexpensive RGB-depth sensors such as Microsoft Kinect [5], many researchers have recently incorporated depth information to solve computer vision problems. Examples include action recognition and object classification [6]. Code libraries for the Kinect sensor [7,8] include algorithms for person detection and skeleton tracking. Using these functionalities, the 3D positions of body joints and the silhouette of the subject can be obtained in real time, allowing researchers to focus on the subsequent processing steps.

Despite the progress in developing action recognition techniques using skeletal data, many of existing approaches are not robust in handling inaccuracies in the tracking of body joint positions, particularly when the 3D information comes from range sensors [9]. Furthermore, researchers commonly use either the orientation of joints or their pairwise positions as action descriptors. Simple actions involving the use of only one body part (e.g., hand waving or brushing teeth) can be identified by considering orientations of the joints of the individual body part. In complex actions involving movements of two or more limbs (e.g., walking or a tennis serve), relative motions of the participating joints associated with the specific action are more discriminative than the individual joint trajectories, as they represent temporal relationships amongst different body limbs. For example, when walking, a person's hands tend to move in opposite directions. Therefore, the pattern of the relative motion of hand joints while walking may be a distinctive feature. This additional cue cannot be captured when the trajectory of each hand is considered separately. In addition, even though complex actions can be performed differently resulting in dissimilar individual joint trajectories, the relative motions of participating joints have consistent characteristics in their pattern.

The goal of the proposed approach in this paper is to employ a combination of spatio-temporal based skeleton features and a new similarity function based on the longest common subsequence (LCSS) algorithm [10] for dealing with both simple and complex actions. A new method for finding discriminative joint information for each type of features for each action class is also proposed here. Using the combination of discriminative features, we classify newly acquired action instances by comparing their similarity to

☆ This paper has been recommended for acceptance by M.T. Sun.

* Corresponding author.
E-mail address: hosseinp@ecu.edu.au (H. Pazhoumand-Dar).

http://dx.doi.org/10.1016/j.jvcir.2015.03.002
1047-3203/© 2015 Elsevier Inc. All rights reserved.

learned actions using the LCSS algorithm [10], which makes our classification invariant, with respect to temporal variations, different subjects, and noisy 3D joint positions. To the best of our knowledge, the LCSS algorithm has mainly been applied to silhouette-based action recognition in 2D video sequences [11,12] but is used here as a similarity function for action recognition based on 3D joint tracking information. LCSS has shown good performance in matching actions using optical flow motion features extracted from video images [12] and unlike dynamic time warping (DTW), it is robust to outliers and noise in the sequences. In contrast to Jin and Choi [13] who used LCSS to identify discriminative joints, the proposed approach used LCSS to classify actions, resulting in a method that is robust to actions performed at varying speeds.

The rest of the paper is organised as follows: Section 2 briefly reviews relevant literature on human action recognition. Section 3 begins by describing our feature vector extraction technique from skeletal data and then details steps of classifying newly acquired action instances. The experimental evaluation of our technique is presented in Section 4 followed by conclusions and future directions in Section 5.

2. Previous work

Visual human action recognition and analysis is a well-established field that employs motion sequences to identify actions found in video. Existing approaches can be coarsely grouped into two classes. The first group employs 2D image sequences, while the other is based on 3D trajectories.

A common trend for 2D action recognition involves using image features to characterise the spatial–temporal structure of actions. For example, the Space–Time Interest Points (STIPs) operator was used by Laptev [14] to detect regions with large local variations in both spatial and time domains. Chung and Liu [15] classified actions by statistically analysing primitive body postures. In another study, Lu and Little [16] calculated descriptors known as Histograms of Oriented Gradients (HOG) to characterise the subject's actions using a probabilistic graphical model.

Video images provide a rich source of information for monitoring a scene; however, they depend on appropriate lighting. In addition, since human motions are performed in three-dimensional space, using 2D video images reduces the discriminative ability of approaches for action recognition. Since the release of inexpensive RGB-depth cameras, such as the Microsoft Kinect, human action related research based on depth information has accelerated. These sensors improve on traditional RGB cameras, by providing depth information and are robust to light, colour, and texture variations. Ni et al. [17] collected a dataset for a health care problem and integrated depth information in two conventional image-based feature descriptors, STIPs and Motion History Images, to describe relevant action classes. Their results illustrate that integrating depth and RGB information produces more inter-class discriminating capability.

Recent depth-based action recognition approaches have incorporated the SVM classifier and new representations for describing actions. Xia and Aggarwal [18] presented an algorithm for extracting STIPs and depth cuboid similarity features (DCSF) from sequences of depth maps for characterising actions. In that approach, first the DCSF extracted from training data are clustered to define a codebook. Afterwards, each sequence of depth maps associated with an action was represented as a "bag-of-codewords." The histograms of these bag-of-codewords were used to train SVMs for classification of unseen action instances. The approach proposed in [19] projected depth maps of action instances onto three orthogonal planes and accumulated motion energy through entire sequences of each projection to generate the Depth Motion Maps (DMM). HOG descriptors were then used to characterise DMMs to form the final representation for the specific action. Yang and Tian proposed another representation for describing actions known as Super Normal Vector [20]. This representation is based on a codebook of polynormals where polynormals were first obtained by concatenating normal vectors associated with local spatio-temporal neighbourhoods of each cloud point in a depth sequence of an action. A sparse coding approach was then employed to compute the polynormal codebook. In another approach, Hadfield and Bowden [21] extended common algorithms for interest point detection, such as Harris [14] and Hessian [22] operators to extract salient features from depth maps. To characterise actions, these authors computed three features, namely, HOG, Histograms of Oriented Flow [14], and Relative Motion Descriptor [23] for each detected interest point in a depth map.

Johansson [24] showed that human observers can recognise actions by observing only the main joints of a human body. Hence, several approaches, listed in Table 1, have employed different types of features extracted from the position of skeleton joints in each video frame for action classification. One of the most commonly used feature is the orientation of joints, which is commonly computed as the internal angle between adjacent joints [13] or the projection angle of joint vectors onto specific planes [25,26]. From skeletal joint locations, Yang and Tian [27] used the relative differences of joint positions, which is a compact representation of the structure of the skeleton for actions involving multiple joints. In a number of existing approaches, the displacement of joint position between the current frame and other frames is calculated in order to capture the joint motion across time [28–30]. There are also some techniques associating features extracted from depth data with 3D joint data for action recognition that involves interaction with the environment [31,32]. Xia et al. [33] proposed a histogram-based representation for human actions, in which a spherical coordinate system is employed for binning the joint positions. After applying Linear Discriminant Analysis (LDA), feature vectors are then clustered into a "K posture" vocabulary to capture the frequency of skeletal joints located in each histogram bin.

A number of other approaches combine different types of joint features [28–30,32] but as all skeleton joints in each frame are involved, noise introduced by unrelated joints during specific actions may degrade the classification result. Furthermore, since they mostly use direct concatenation of relative positions of joints in combination with joint displacements across time, noise introduced in one feature will impact the discriminative power of the entire feature vector.

Recently, several algorithms have been proposed for identifying related joints during actions. For example, Jin and Choi [13] employed the LCSS algorithm to find essential joints with a number of shared symbols in their feature trajectories during each action. Wang [31] trained Support Vector Machine (SVM) models on

Table 1
Skeletal joint features used in different action recognition techniques.

Method	Relative positions of joints	3D position of joints	Joint orientations	Joint displacements across time	Depth features
[13]			√		
[27]	√			√	
[28]	√		√	√	
[29]	√			√	
[30]	√			√	
[31]	√				√
[32]			√	√	√
[33]			√		
[34]		√			
[35]			√		

feature vectors of each joint for each action. In their approach, a joint is considered informative if its associated SVM model gives a high probability to the samples of the same action and a low probability to those of other actions. Using a weighting scheme in DTW cost computation, Reyes et al. [36] proposed a technique to assign weights to each body joint based on their contributions to all learned action classes. Since each body joint is associated with a single weight for all action classes, unrelated joint information may be included in feature extraction.

To model features extracted from 3D joint positions, many different approaches have been proposed. One way to model the human actions is to employ generative models, such as a Hidden Markov models (HMMs) for a number of pre-defined postures [37], context-free grammar [34], or finite state machines for sequences of joint orientations [13]. Since estimated skeletal joint positions are generally noisy, a problem with these techniques is the difficulty of correctly determining states in cases of candidate actions with small differences.

Approaches based on explicitly learning the spatio-temporal patterns of the joints are also proposed for skeleton-based action recognition. Ofli [38] and Wang [31] presented a way for classifying joint features using SVM. Adaptive Boosting (AdaBoost) is used as a classifier by Bloom et al. [28]. However, with a limited amount of training data, such models are easy to overfit.

DTW on the other hand does not require training data as it finds the best alignment between 3D joint trajectories of new instances and learned templates with different lengths [35,36,39–41], and therefore, action recognition can be done through a nearest-neighbour classification scheme. Nevertheless, in DTW, outliers in the estimated positions of occluded joints can distort the distance function and consequently, degrade the classification accuracy. Hence, new techniques that are robust to noisy or missing joint positions would be valuable.

3. The proposed approach

The proposed approach in this paper consists of two phases: training and classification. Fig. 1 shows the steps in the training phase, namely, data representation and feature vector selection. The data consists of a number of action classes and in each action class the subject performs a number of action instances. In the data representation step, positions of skeleton joints associated with a tracked subject are first captured and followed by some pre-processing operations. Next, in the feature vector selection step, two descriptors are obtained: sequences of joint angles and relative positions of joint pairs. Then, using these action descriptors, two features, namely, *the most informative sequences of joint angles* (*MIJA*) and *the most informative relative motions* (*MIRM*) are obtained to characterise each action class. Both these two features allow us to extract the discriminative sets of participating joints associated with each action class. Details associated with these steps are outlined in Section 3.1.

In the classification phase, newly acquired action instances are first similarly processed for extracting the features. The classification approach is described in Section 3.2.

3.1. Training phase

This section details the data representation and feature vector selection steps in the training phase of the proposed approach. Section 3.1.1 described the technique used to extract the skeletal representation employed in this study. This is followed by Section 3.1.2 which describes the approach to generate sets of the most discriminative joints that characterised each action class using the output from the data representation step.

3.1.1. Skeletal representation

The current version of the Kinect SDK [7] provides 3D positions of 20 skeletal joints for each subject tracked in the scene. As shown in Fig. 2(a), these points form a skeletal representation of the human pose; each of them has an associated state (e.g., tracked, not tracked, or inferred) indicating the tracking status. In our approach, we consider the position of major joints associated with movement of hands, feet, and the head of the skeletal structure, as we found them to be more relevant in recognising human actions from our experiments. We choose to use 11 out of the 20 joints, comprising of the *right foot, left foot, right knee, left knee, right wrist, left wrist, right elbow, left elbow, right hand, left hand,* and *head*, for

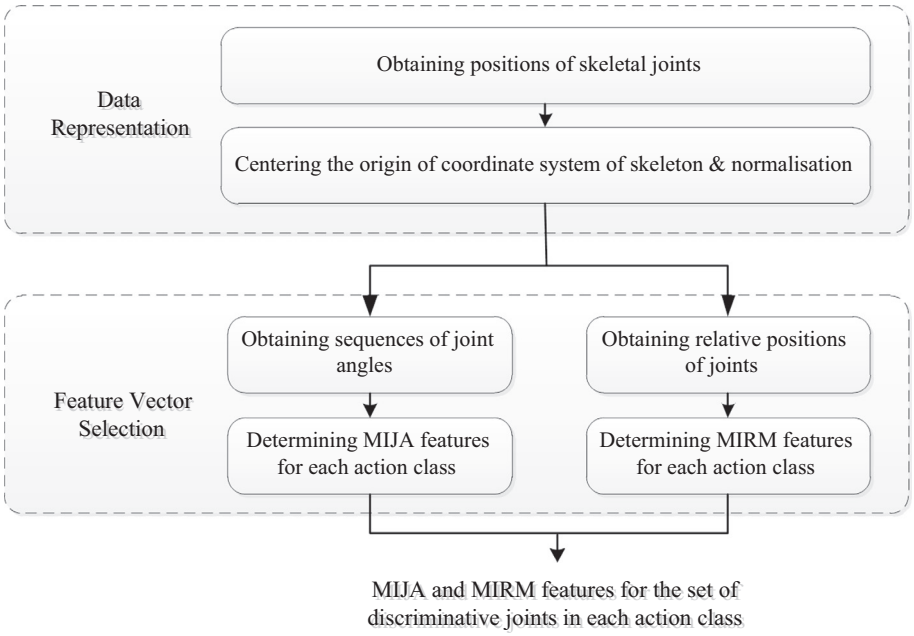

Fig. 1. Steps of the training phase of the proposed approach.

capturing information for an "action." For skeletal data captured from other motion recording systems such as mocap, we consider similar corresponding joints to those we have chosen for the Kinect skeletal representation.

After the 3D positions for joints of interest are captured, pre-processing operations are applied on them. First, we center the skeleton, using the hip center joint as the origin (0, 0, 0) of the coordinate system. From this origin, vectors to the 3D position of each joint j_i are calculated using the approach proposed by [25,26]. Let $\vec{f}_{j_i}^{t} \in R^3$ be the 3D location $\left(x_{j_i}^t, y_{j_i}^t, z_{j_i}^t\right)$ of joint j_i, $i = 1 \ldots 11$, at time t. The resulting body-joint vectors $\vec{f}'t_{j_i}$ represent the 3D structure of the human body at time t via the orientation of joints. An example is shown in Fig. 2(b).

In the next step of pre-processing, Z-score normalisation [42] is performed for each subject separately to make our representation robust to different body sizes. That is, we normalise 3D location of each joint j_i of each subject at time t using mean vector $\vec{f}_{j_i}^{*}$ and standard deviation value $\sigma_{\vec{f}'_{j_i}}$ of that joint, calculated across all action instances performed by the subject (see Eq. (1)).

$$\vec{f}_{j_i}^{t} = \frac{\vec{f}_{j_i}^{\prime t} - \vec{f}_{j_i}^{*}}{\sigma_{\vec{f}'_{j_i}}} \tag{1}$$

3.1.2. Feature vector selection

This section describes techniques, employed in the second step of the training phase, to describe actions using normalised 3D location of body joints obtained in the first step. Two descriptors are extracted to characterise actions: (1) sequences of joint angles to characterise body poses and (2) the relative motions of joints, to describe the temporal relationships between involved joints, which is especially meaningful for complex actions. Furthermore, since during different actions, different sets of joints are involved, an algorithm for each of these two descriptors is developed for selecting the most discriminative joint information for each action class, namely, *MIJA* and *MIRM*. For example, to calculate the *MIJA* features for action class "*hand waving,*"three joints of the involved hand may be selected by the algorithm whereas the joints in both hands and feet may be chosen for the action class "*walking.*"

3.1.2.1. The most informative sequences of joint angles (MIJA). The shape of trajectories of joints encodes local motion patterns for each action. Let $F_{j_i} = \{\vec{f}_{j_i}^{t}\}_{t=1}^{t=e}$ represent vectors from the origin to a joint's positions in an action sequence, where e is the number of frames in the specific action sequence. Since different individuals tend to perform actions with different speeds, the length of each instance of an action can be different. Hence, e is a variable that captures the different numbers of frames associated with different sequences and is used in subsequent processing to extract features.

Similar to techniques in [43,44], for each vector in F_{j_i}, we calculate θ as the angle of its projection onto the x–z plane with the positive x axis. Also, φ is captured as the angle between the vector and the y-axis (see Fig. 3). These values can uniquely capture the orientation of any vector.

Hence, $D(j_i) = \left\{(\theta^t, \varphi^t)\right\}_{t=1}^{t=e}$ is a sequence related to the values associated with different angles of joint j_i during an action. During our experiments, we observed that for different actions, different numbers of joints were involved. As a result, if we consider a fixed number of joints (e.g., all possible joints or a finite number of them), features of those joints that might not be active in characterising some actions are being incorporated into the classification stage. For instance, the movement of the legs when drawing a circle by hand should not have an influence on classification and, thus, using them will add noise to this stage. In this sense, we propose calculating vectors $D(j_i)$ for only those joints that have high contributions for the specific action classes. More precisely, the level of engagement of each joint j_i in the action class k is estimated based on the degree of variation in its position using the theory of information entropy [45]. If θ and φ for each joint j_i during all instances of the action class k have N and M different angle values with the probabilities $\{p_n^{\theta}, n = 1 \ldots N\}$ and

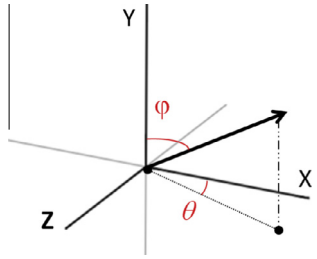

Fig. 3. Calculation of θ and φ for a joint vector.

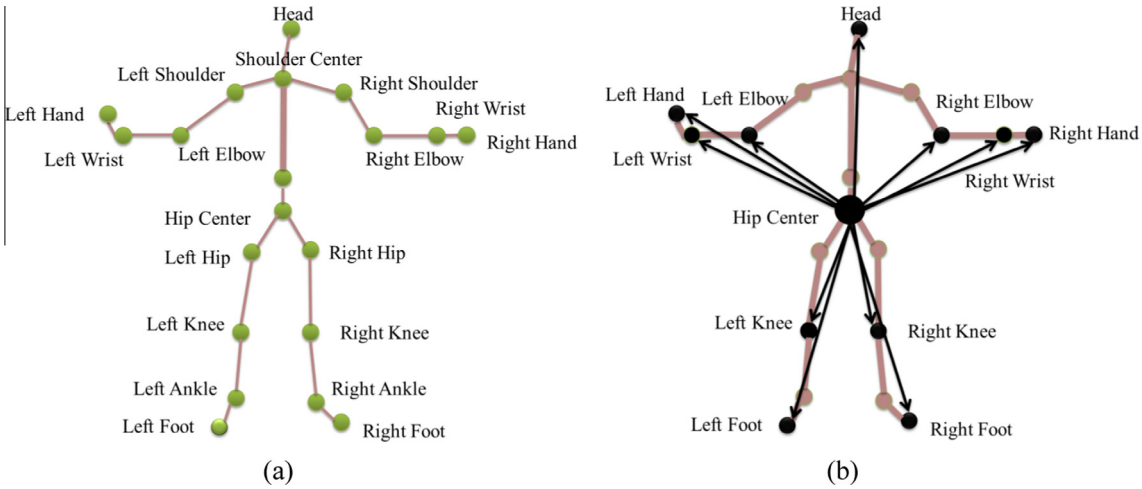

Fig. 2. (a) Kinect skeletal joints representation and (b) the vectors from the origin to the joints of interest. Each vector is used to represent the orientation of the joint.

$\{p_m^{\varphi}, m = 1 \ldots M\}$respectively, then the entropy of those joint angles for joint j_i is calculated as follows:

$$H_{j_i} = -\sum_{\text{for all samples of the action class } k} \left(\sum_{n=1}^{N} p_n^{\theta} \log_2 p_n^{\theta} + \sum_{m=1}^{M} p_m^{\varphi} \log_2 p_m^{\varphi} \right) \quad (2)$$

As entropy is a measure of uncertainty in a variable, its quantity in our case indicates the level of spread for different angle values associated with a joint. Since our objective is to identify *MIJA* for each action class, logically we need to use a procedure to obtain a threshold value (H') to filter out joints that have a low entropy (i.e. they are not participating in the specific action classes and therefore, their H_{j_i} is lower than H'). Here, we use a data driven approach in which two fold cross validation is performed where half of subjects in the training data are used as a training set and the accuracy of the algorithm is validated using the remaining half of the training data. In addition, the cross subject validation technique is also employed and the results are then averaged. For each of the two validation runs, we vary the value of H' to obtain the highest accuracy rate of the algorithm on the validation set. Using this threshold value, each action class can be associated with a different set of essential joints.

3.1.2.2. The most informative relative motions of joints (MIRM). In order to describe the characteristic aspects of complex actions involving more than one body limb (i.e., hand clap, tennis swing, and walking), we employ a feature that represents discriminative simultaneous or consecutive relative motion of joints during actions. That is, considering two joints j_x and j_y with the feature vectors F_{j_x} and F_{j_y}, the sequence of their positions relative to each other during an action (i.e. distance between joints) is calculated as Eq. (3).

$$SRP(j_x, j_y) = \left\{ \left\| \vec{f}_{j_x}^{t} - \vec{f}_{j_y}^{t} \right\| \right\}_{t=1}^{t=e} \quad (3)$$

Fig. 4 shows $SRP(j_{left\ hand}, j_{right\ hand})$ for three action classes, namely, "*jogging*," "*a tennis serve*," and "*a two hand wave*." Each graph shows two instances of the specific action class, performed by two different subjects. While the patterns in the sequences belonging to different action classes are distinguishable, for a specific action class, they show some similar characteristics between instances for the same action but performed by different subjects.

To learn the *MIRM* for each action class, an algorithm (as shown in Fig. 5) is developed in which the inter and intra action variations for the differences in 3D positions of each joint pair are considered. To be more specific, given a number of samples for P action classes, the algorithm extracts joint pairs that have the *MIRM* for each action class. In line (4) and (5), the variance is calculated across the $SRP(j_x, j_y)$ during all instances of the current action class (a) and other action classes, respectively. Consequently, each element (x, y) in the matrix D_{within} shows the mean value of variations in relative positions of joints x and y for the current action class. Likewise, that element in the $D_{between}$ shows the same metric for all samples of other action classes, excluding the current action class. As the output, the index of joint pairs where their D_{within} is greater than their $D_{between}$ is returned for each action class in line 11 of Fig. 5.

In classical object recognition problems, a descriptor is more discriminative when its inter-class variation is greater than its intra-class variation. However, for complex actions involving more than one body limb, the variation in relative positions of joint pairs that are involved during an action is higher than other joints. Therefore, for instances of such action classes, only those joint pairs whose intra-class variation (D_{within}) is higher than their inter-class variation ($D_{between}$) can have a discriminative pattern in their relative positions. Note that, for simple actions performed by only one body limb, the algorithm might return no joint pair and thus, those actions will be classified based on only *MIJA* features.

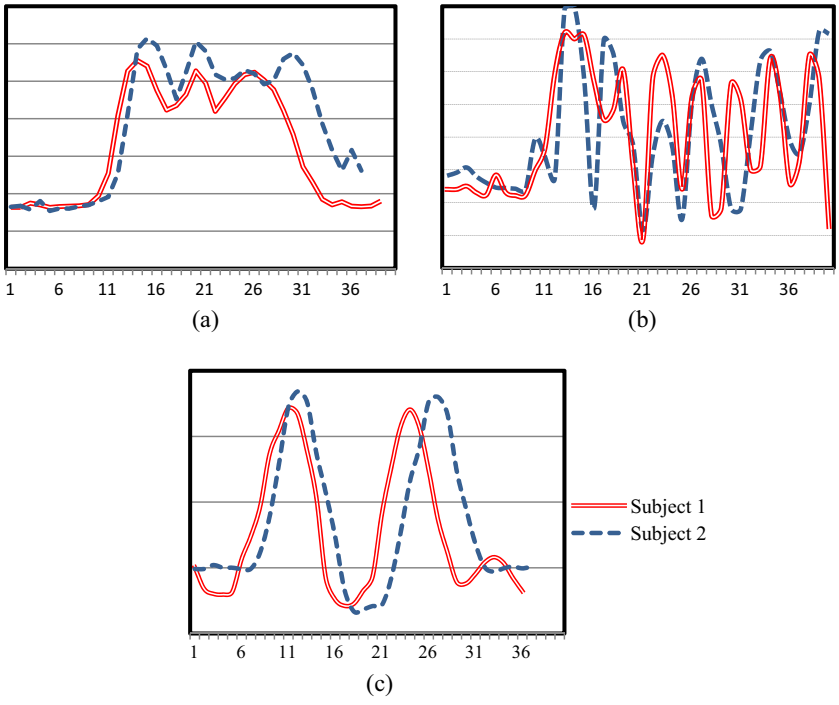

Fig. 4. $SRP(j_{left\ hand}, j_{right\ hand})$ during action classes (a) *jogging*, (b) *tennis serve*, and (c) *two hand wave*, performed by two subjects. Horizontal scale shows the frame number and the vertical scale shows the relative distance.

Identifying the MIRM for each action class
Input: *training instances for all P action classes*
Output: *set of informative joint-pairs for each action class*

1. $For\ a = 1{:}P\ Do$ $//number\ of\ action\ classes$
2. $\quad For\ x = 1{:}N\ Do$ $//number\ of\ joints$
3. $\quad\quad For\ y = x+1{:}N\ Do$
4. $$D_{within}(x,y) = Mean\left(Var\left(SRP\left(j_x, j_y\right)\right)_{for\ all\ samples\ of\ action\ classes\ (a)}\right)$$
5. $$D_{between}(x,y) = Mean\left(Var\left(SRP\left(j_x, j_y\right)\right)_{for\ all\ samples\ of\ other\ action\ classes, excluding\ class\ (a)}\right)$$
6. $$V_{(x,y,a)} = Max\left(0, \frac{D_{within}(x,y) - D_{between}(x,y)}{D_{within}(x,y)}\right)$$
7. $\quad\quad End\ for$
8. $\quad End\ for$
9. $End\ for$
10. $For\ a = 1{:}P\ Do$ $//number\ of\ action\ classes$
11. $\quad Return\ the\ index\ (x,y)\ of\ nonzero\ values\ in\ V_{(x,y,a)}$
12. $End\ for$

Fig. 5. The algorithm for identifying joint pairs with discriminative relative motion.

3.2. Classification phase

This section addresses the classification phase of the proposed approach. We process newly acquired action instances in the same manner as described in Section 3.1.1. Next, *MIJA* and *MIRM* features for each of the newly acquired data instances are calculated against each known action class using its corresponding discriminative set of participating joints obtained in the training phase. In order to classify these instances, we evaluate their similarity to the training instances in specific action classes using the approaches outlined in Sections 3.2.1 and 3.2.2.

3.2.1. Similarity measure

A common way of comparing and classifying features extracted from joint trajectories is through using classifiers such as SVM, decision tree, or naive Bayes. This requires performing pre-processing operations to normalise the length of action sequences which have different frame numbers or different speeds [34]. In addition, if such techniques use the Euclidean distance, outliers in joint trajectories can result in considerably large dissimilarities between extracted features, thus often necessitating the need to remove outliers. To overcome these limitations, we propose to use a similarity function based on the non-metric LCSS [10] for a number of reasons, mainly because it is an effective technique for comparing similarity between sequences which may be "*out of phase*" and are of different lengths. It is also robust against noise and outliers in the sequences. Unlike SVM, it handles multiclass classification better from the perspective that SVM is a binary classifier with high algorithmic complexity and extensive memory requirements. To handle multiclass classification using SVM, work arounds such as pair-wise classifications involving one class against other classes and then the step repeated for all classes have to be carried out. In addition, while SVM classifiers can produce very accurate results, parameters associated with SVM such as the choice of kernel and kernel parameters have to be chosen appropriately. This leads to situations where a set of parameters can produce excellent classification accuracy for one problem and yet produce poor results when applied to another problem. Thus, parameter tuning often needs to be carried out to find the set of parameter settings for achieving satisfactory classification results. In addition, LCSS is an algorithm that is robust to different spatial locations of the specific body parts involved in an action sequence [10]. For instance, in the case of "hand waving," one action sequence involves a subject waving his/her hand in front of his/her chest, while in another sample, the hand waving may be performed when the hand is above the head. Fig. 6 illustrates an example of matching two action trajectories in 2D space. Note that the alignments for measuring the distance between the two sequences are only performed for the points that have a distance below a threshold value in both time and space. Hence, as outliers are not matched, this similarity measure offers enhanced robustness under noisy conditions.

For the *MIJA* features, the distance between two sequences of joint orientations is calculated as a pair of angles using a non-metric space. First, in order to eliminate the small deviations across instances of the same class of action and to reduce the computational cost, a uniform quantization to 36 angles procedure is performed on θ and φ. Then, the difference of two points in the sequences is represented as the distance between their

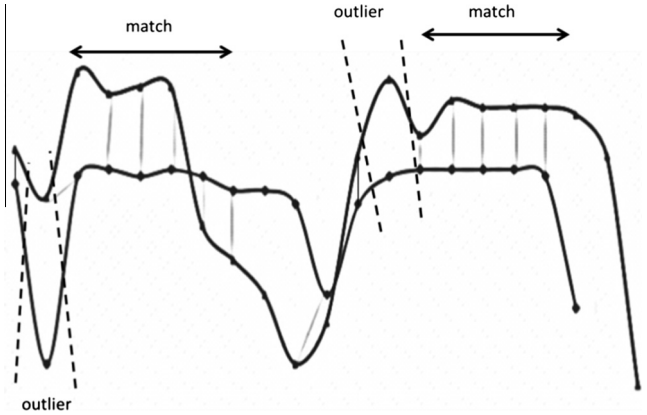

Fig. 6. Matching of two trajectories with outliers using LCSS.

representative angles. Specifically, given $D(j_i) = \{(\theta^t, \varphi^t)\}_{t=1}^{t=e}$ as a sequence of joint angles with the size e, we define $Head(D(j_i))$ as described in Eq. (4).

$$Head(D(j_i)) = \{(\theta^t, \varphi^t)\}_{t=1}^{t=e-1} \tag{4}$$

By choosing integer δ, and a real number ∂_ε, the function $LCSS_{\delta,\partial_\varepsilon}$ for two sequences of joint angles, $D_1(j_i) = \{(\theta_1^t, \varphi_1^t)\}_{t=1}^{t=e_1}$ and $D_2(j_i) = \{(\theta_2^t, \varphi_2^t)\}_{t=1}^{t=e_2}$ is defined as Eq. (5). Note that the LCSS algorithm is designed to compare similarities between sequences with different lengths. In Eq. (5), e_1 and e_2 are the number of points in each sequence of joint angles, respectively. These determine the initial point from which $LCSS_{\delta,\partial_\varepsilon}$ exhaustively tries to match data between the two sequences. Finally, Eq. (5) returns the number of matched points in the two sequences. For two sequences of different lengths, the theoretical minimum output from LCSS is 0 (no points matched) and the theoretical maximum is the length of the shorter sequence (all points in the shorter sequence matched to points in the longer). Hence, as the role of "e" is of the nature of an indexing element, there is no need to normalise "e". For more detail description we refer the reader to [10].

$$LCSS_{\delta,\partial_\varepsilon}(D_1(j_i)D_2(j_i)) = \begin{cases} 0 & \text{, if } D_1(j_i) \text{ or } D_2(j_i) \text{ is empty} \\ 1 + LCSS_{\delta,\partial_\varepsilon}(Head(D_1(j_i)), Head(D_2(j_i))), & \text{if } |e_1 - e_2| < \delta \text{ and } |\theta_1^{e_1} - \theta_2^{e_2}| < \partial_\varepsilon \text{ and } |\varphi_1^{e_1} - \varphi_2^{e_2}| < \partial_\varepsilon \\ Max(LCSS_{\delta,\partial_\varepsilon}(Head(D_1(j_i)), D_2(j_i)), LCSS_{\delta,\partial_\varepsilon}(D_1(j_i), Head(D_2(j_i)))), & \text{otherwise} \end{cases} \tag{5}$$

In Eq. (5), ∂_ε limits the maximum distance between two angles in the sequences. Hence, as the outliers make sudden changes in trajectory values, the LCSS algorithm only matches two corresponding points in two sequences if their angular difference is below ∂_ε. Large values for this parameter result in dissimilar sequences being matched as "similar" by the LCSS, whereas too small values make the matching more specific (i.e. only sequences that follow the exact same pattern are matched). For this parameter, we use the smallest standard deviation by two examining sequences as suggested by [10]. Therefore, ∂_ε might vary for each pair of sequences being compared during the process, depending on the value of their standard deviation.

The speed of performing action instances can also be varied depending on the subject performing the action. However, it is logical to put bounds on the expected speed an action instance. δ in Eq. (5) controls how far in time we can go forward or back in order to match a given point from one trajectory to a point in another one. We experimentally determined this parameter to 15% of the shorter trajectory.

In order to detect similarities in movement of joints with different displacement in each dimension, we define a set of translation functions $T_{\alpha,\beta}$ (as shown in Eq. (6)) that allows us to move sequences in the angular space.

$$T_{\alpha,\beta}(D(j_i)) = \{(\theta^t + \alpha, \varphi^t + \beta)\}_{t=1}^{t=e} \tag{6}$$

Given a number of translation functions, the similarity function between two sequences $D_1(j_i)$ and $D_2(j_i)$ is calculated as follow:

$$Similarity_{MIJA}(D_1(j_i), D_2(j_i)) = Max\left(\frac{LCSS_{\delta,\partial_\varepsilon}(D_1(j_i), T_{\alpha,\beta}(D_2(j_i)))}{Min(|D_1(j_i)|, |D_2(j_i)|)}\right) \tag{7}$$

In Eq. (7), different translation functions are used to compare the two sequences having different lengths. In this calculation for similarity, the issue of differences in length of the two sequences is then addressed specifically in the denominator in Eq. (7). The number of matched points for each comparison is divided by the length of the shorter sequence, resulting in a value between 0 and 1. Maximization is then performed over results.

3.2.2. Classification algorithm

Fig. 7 shows the algorithm used for classification of new data instances. Let T be the set of all training samples in which each one corresponds to an action instance of known action classes in the dataset. We partition T to P disjoint subsets $(T_1, T_2, \ldots, T_P)$ such that T_a contains all instances of action class a ($1 \leqslant a \leqslant P$). Given a test action instance (Q), the *MIRM* and *MIJA* feature vectors for the participating joints of a known action class (a) are calculated and stored as $D(j_i)_Q$ and $SRP(j_x, j_y)_Q$, respectively. Afterwards, the mean distances in matching $D(j)_Q$ and $SRP(j_x, j_y)_Q$ against the same feature vectors associated to each instance in T_a are computed and stored in $Dist_{MIRM}(a)$ and $Dist_{MIJA}(a)$, as shown in lines 3 and 5, respectively. This operation is repeated for all P known action classes in the dataset and then, a minimum-distance classifier finds the most similar action class, based on the combination of distances for each type of feature (line 7).

Note that here the same method we described in Section 3.2.1 is used to calculate the $Similarity_{MIRM}(.,.)$. The two inputs to this function are obtained from Eq. (3) and similar to $Similarity_{MIJA}(.,.)$, the output ranges from 0 to 1.

4. Results and discussion

We evaluate the performance of our action recognition approach using real-world examples. We first describe the action datasets used to evaluate our approach and the results of learning the parameter value to obtain *MIJA* features for each given dataset. Next, we demonstrate the value of combining both feature types by comparing the accuracy of the approach using only one type of feature against that of using the combination of features as well as against other existing approaches. Finally, we provide a comparison between the robustness of our classifier to that of the other most similar template matching techniques used in the literature.

4.1. Datasets

We evaluate the performance of the proposed action recognition technique on two different publicly available datasets of skeletal data corresponding to human actions, namely MSR Action3D [31] and HDM05 [46].

The MSR Action3D dataset was captured by a Kinect camera at a frame rate of 15 frames per second; it contains 20 action classes performed by ten subjects. Various movements of single or multiple body limbs are covered in this dataset including *arm wave, two hand wave, side kick, jogging*, etc. These action classes were chosen to capture single or combinational motions related to arms and legs. In the case of simple actions, subjects used their right hand or leg. All ten subjects performed each action three times resulting 600 samples. According to [31], some samples have missing skeleton data and excluding them leaves 557 sequences for our experiments. In order to compare with a number of existing methods,

Classification algorithm
Input: *the query sequence (Q) of skeleton joint data*
Output: *the most similar action class index (m)*

1. *For* $a = 1{:}P$ *Do //number of action classes*
2. *Compute* $D(j_i)_Q$ *and* $D(j_i)_a$ *as MIJA features for Q and all members of* T_a*, respectively where* $j_i \epsilon$ *{informative joints during action class (a)}*
3. $$Dist_{MIJA}(a) = Mean\left(\sum_{for\ all\ members\ of\ T_a} 1 - Similarity_{MIJA}\left(D(j_i)_Q, D(j_i)_a\right)\right)$$
4. *Compute* $SRP(j_x, j_y)_Q$ *and* $SRP(j_x, j_y)_a$ *as MIRM features for Q and all members of* T_a*, respectively where* $(j_x, j_y) \epsilon$ *{informative joint pairs during action class (a)}*
5. $$Dist_{MIRM}(a) = Mean\left(\sum_{for\ all\ members\ of\ T_a} 1 - Similarity_{MIRM}\left(SRP(j_x, j_y)_Q, SRP(j_x, j_y)_a\right)\right)$$
6. *End for*
7. $m = arg\min_{a=1:P}\left(Dist_{MIRM}(a) + Dist_{MIJA}(a)\right)$
8. *Return m*

Fig. 7. The algorithm for classification of a test action instance using the minimum-distance classifier.

we used only one combination involving samples of Subjects 1, 3, 5, 7 and 9 as training dataset and those of Subjects 2, 4, 6, 8 and 10 forming the unseen test dataset. This partitioning of the MSR Action3D dataset into the training and unseen test dataset has been used in many depth-based approaches and is outlined in [31].

HDM05 motion capture (mocap) dataset contains several skeleton motion data belonging to more than 70 action classes performed by five subjects with different numbers of repetition. Each sequence was captured using a motion capture system at a rate of 30 frames per second. As samples belonging to different action classes can be very similar, this dataset is very challenging. We used the same action classes and test setting as in [38], where three subjects were used for training and the others for testing, resulting in 250 mocap sequences, as described in Table 2. There are 31 recorded joints in the subject's skeletal model and, thus, we considered 11 joints similar to those explained in Section 3.1.1.

4.2. Experimental results on action classification

In the first experiment, the approach described in Section 3.1.2.1 was employed on both datasets to learn the value of H' for each dataset respectively. After each iteration of the twofold cross validation stage, the accuracy for each iteration is obtained using Eq. (8),

$$Accuracy = \frac{TP}{\sum_{r=1}^{P}\sum_{c=1}^{P} C(r,c)} \quad (8)$$

where *TP* is the number of correctly labelled instances and *P* is the number of action classes. Fig. 8 shows the results for the HDM05 dataset. From the training it can be seen that the value for this parameter is 0.3.

Accordingly, the corresponding sets of joints for each action class were identified as shown in Fig. 9. Each column shows the set of the most involved joints for the corresponding action. It can be observed that some actions such as "*action ID 4*" and "*action ID 8*" involve the use of few joints (i.e., right wrist, left wrist, right hand, and left hand), while some other actions like "*action ID 9*" involve movement of more joints (i.e., right wrist, left wrist, right hand, and left hand, right foot, left foot, and left elbow).

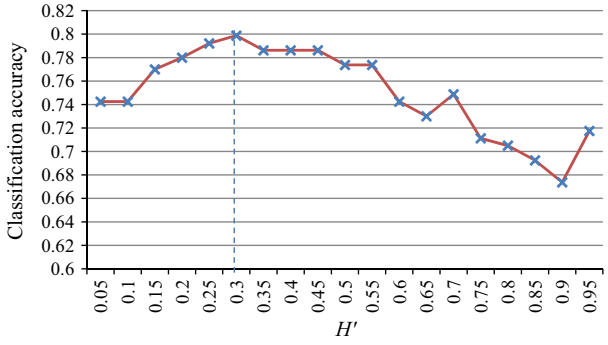

Fig. 8. The average classification accuracy resulting from different values for H' for the HDM05 dataset.

Table 2
Action classes used from HDM05 mocap dataset.

Action ID	Action class	Number of sequences	Number of frames
1	Deposit floor	32	11,623
2	Elbow to knee	13	5756
3	Grab high	29	7506
4	Hop both legs	12	2327
5	Jog	17	4142
6	Kick forward	29	6225
7	Lie down floor	20	13,100
8	Rotate both arms backward	16	1742
9	Sneak	16	3770
10	Squat	52	10,035
11	Throw basketball	14	5710

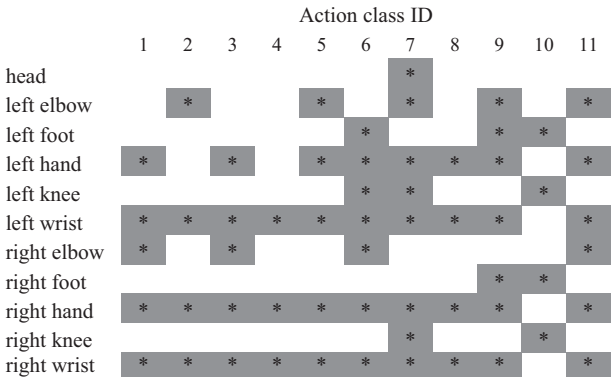

Fig. 9. Corresponding set of joints identified for computing *MIJA* features in the HDM05 dataset.

For the MSR Action3D dataset, $H' = 0.3$ was also obtained from the training phase. Next, the algorithm proposed in Section 3.1.2.2 was applied on each of the two datasets to identify the joint pairs with *MIRM* for each action class. Fig. 10(a)–(e) shows some examples from the MSR Action3D dataset of the set of joints and joint pairs used for obtaining the *MIJA* and *MIRM* for action classes for "*tennis serve*", "*side boxing*", "*jogging*", "*pick up and throw*", and "*horizontal arm wave*", respectively. It can be observed that action classes with different complexity result in different sets of *MIJA* and *MIRM* features. For the sake of brevity, only five action classes are illustrated from this dataset in Fig. 10.

In real life, when a person is performing a tennis serve, most times the joints associated with both hands are actively moving. In addition, the relative position pattern of both hands is unique for that specific action class. The results from our experiments as presented in Fig. 10(a) conforms this assumption. As seen, joints of the right and left hands shown as a red circle indicating that both hands are used to calculate *MIJA* and the *MIRM* of hands (indicated by the dotted line) is also involved. Similarly, when a person is doing side boxing, the same joints are involved but, the hands commonly move in different directions making the *MIRM* involving the right hand and the left elbow more discriminative than the other joint pairs. As seen in Fig. 10(b), for this action class, the algorithm adopted the same set of joints as for the "*tennis serve*" for obtaining *MIJA* but as indicated by the dotted line, *MIRM* is calculated for the right hand and the left elbow. Also, for the action jogging, although all the joints are involved, our experiment detected only seven of them as being essential shown in Fig. 10(c). During the action pick up and throw, the right hand usually reaches the floor and then moves towards the top of the head to throw the object. As Fig. 10(d) illustrates, the technique accordingly adopted the pair of right hand and head joints (indicated by the dotted line) to calculate the *MIRM* feature. For horizontal arm wave however (Fig. 10(e)), our approach did not pick any joint pair for *MIRM* feature and hence, this action is classified based on only *MIJA* features.

In the next experiment, we tested our algorithm by using the determined value of H' and classifying data obtained from the unseen subjects from each dataset. As shown in Table 3, for the MSR Action3D dataset, the most challenging action classes are *bend*, *hand clap*, and *draw X* which have been misclassified as other actions due to their similar variations during execution. Also, "*tennis swing*" is occasionally classified as "*tennis serve*" and "*forward kick*," which is logical since all these actions follow a similar pattern in movements of the right hand. It can be seen from the diagonal elements in Table 3 that amongst the 20 action classes, 15 action classes have the correct classification rate of more than 85%, while the rate for six classes was 100%.

Table 4 compares the approach proposed in this study with existing depth-based and skeleton-based approaches that had been evaluated using the MSR Action3D dataset. All the depth-based approaches in Table 4 used the associated sequences of depth maps in the dataset to classify actions whereas our approach employed the associated skeletal information for each action sequence in the dataset. As the comparisons with depth-based approaches show, using the same combination of subjects in the training and unseen test datasets, our algorithm outperforms the state-of-the-art depth-based algorithms [18,47–50], except for that in [20]. However, a point to note is that all the depth images in the MSR Action3D dataset are "*clean*" and the subjects appear at the same depth to the camera with no objects in background. Our approach used skeletal images and the advantage of the proposed approach in using skeletal joint features over depth-based approaches such as the approach in [20] is that our approach is view invariant (both to the camera locations and subject appearances).

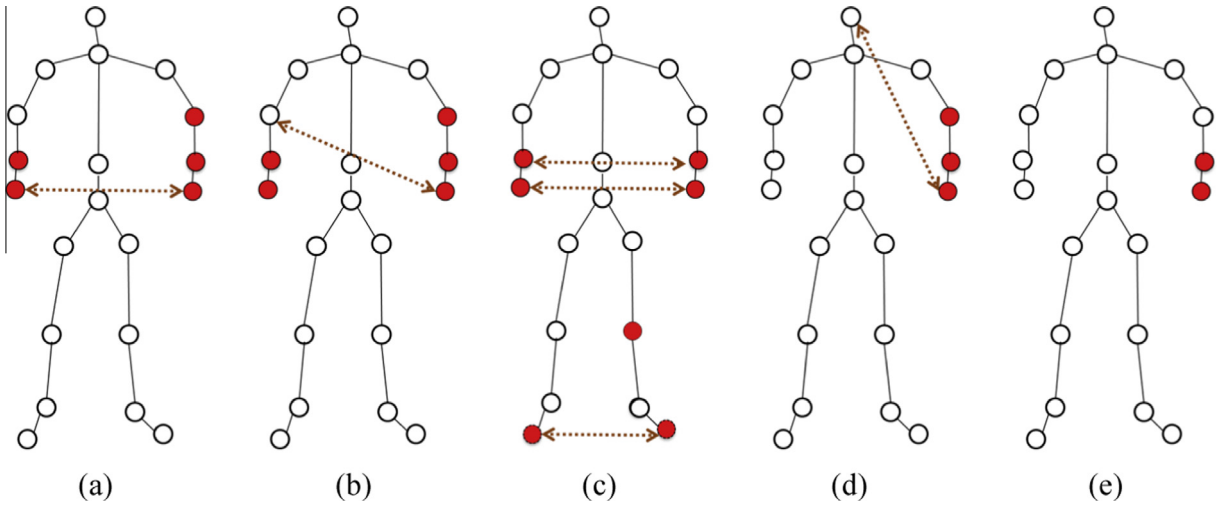

Fig. 10. Results of obtaining *MIJA* and *MIRM* features for action classes (a) "*tennis serve*," (b) "*side boxing*," (c) "*jogging*," (d) "*pick up and throw*," and (e) "*horizontal arm wave*" in the MSR Action3D dataset. Each body joint indicated as a red circle is used for obtaining the *MIJA* feature, and the dotted lines connect joints whose relative position correspond to a *MIRM* feature. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 3
Confusion matrix for action classes in the MSR Action3D dataset by using $H' = 0.3$.

	High arm wave	Horizontal wave	Hammer	Hand catch	Forward punch	High throw	Draw x	Draw tick	Draw circle	Hand clap	Two hand wave	Side boxing	Bend	Forward kick	Side kick	Jogging	Tennis swing	Tennis serve	Golf swing	Pick up and throw
High arm wave	90%	0	0	0	0	0	0	0	0	0	10%	0	0	0	0	0	0	0	0	0
Horizontal wave	0	90%	0	10%	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Hammer	0	0	90%	0	0	0	0	0	0	0	0	0	0	0	0	0	10%	0	0	0
Hand catch	0	0	0	90%	0	0	0	0	10%	0	0	0	0	0	0	0	0	0	0	0
Forward punch	0	0	0	0	100%	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
High throw	0	0	0	10%	0	90%	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Draw x	0	0	0	20%	0	0	80%	0	0	0	0	0	0	0	0	0	0	0	0	0
Draw tick	0	0	0	0	0	0	0	100%	0	0	0	0	0	0	0	0	0	0	0	0
Draw circle	0	0	0	0	0	0	0	0	100%	0	0	0	0	0	0	0	0	0	0	0
Hand clap	0	0	0	0	0	0	0	0	20%	80%	0	0	0	0	0	0	0	0	0	0
Two hand wave	0	0	0	0	0	0	0	0	0	0	90%	0	0	0	0	0	0	0	0	0
Side boxing	0	0	0	0	0	0	0	0	0	0	0	100%	0	0	0	0	0	0	0	0
Bend	0	0	0	0	0	0	0	0	0	0	0	0	80%	10%	0	0	0	0	0	0
Forward kick	0	0	0	0	0	0	0	0	0	0	0	0	0	100%	0	0	0	0	0	0
Side kick	0	0	0	0	0	0	0	0	0	0	0	0	0	10%	90%	0	0	0	0	0
Jogging	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100%	0	0	0	0
Tennis swing	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10%	0	80%	10%	0	0
Tennis serve	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10%	0	0	90%	0	0
Golf swing	0	0	0	0	0	0	0	0	0	0	0	0	0	10%	0	0	10%	0	80%	0
Pick up and throw	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10%	0	0	0	90%

The results of comparisons with state-of-the-art skeleton-based approaches [27,31,33,36,51,52] also show that the proposed approach significantly outperforms other existing skeleton-based methods. The classification accuracy of the techniques HOJ3D [33], EigenJoints [27] and Pose-based Action Recognition [20] are 79%, 82.3% and 90% respectively.

The action-let ensemble model proposed in [31] employs discriminative relative positions of joints extracted from the skeleton, a set of features which is similar to *MIRM* features in this paper. The other feature used in [31] is, however, the Local Occupancy Pattern (LOP) extracted from the depth information around each joint to characterise human-object interactions. Our proposed approach, in contrast, uses *MIJA*, another type of feature extracted from joint trajectories. In the event of where there is no human-object interaction, the approach in [31] can only use one type of feature during classification. For example in terms of the MSR Action3D dataset, the authors reported that they only used the skeleton information because there is no human-object interaction in this dataset. However, our proposed approach learns from both types of features (i.e. *MIJA* and *MIRM*) obtained from the skeletal data, resulting in a higher classification rate for this dataset. In our approach, the use of LCSS allows us to directly capture the temporal structure of actions using the two types of features. This is also in oppose to the technique in [31] where Fourier temporal pyramid was used to implicitly capture the temporal dynamic of actions associated with these features. The disadvantage of using Fourier temporal pyramid is the extra computational cost for obtaining Fourier coefficients. Table 5 summarised the characteristics of these two approaches.

To draw a comparison between LCSS and DTW algorithm, we implemented the method proposed in [36] and evaluated it against the MSR Action3D dataset. As seen in Table 4, the accuracy of the proposed algorithm outperforms the DTW approach in [36] since the LCSS is more resilient to inconsistencies in the joint positions, particularly in case involving occlusions. More specifically, it compares the sequences of actions with different lengths allowing some outliers to be unmatched, whereas in the other technique, like DTW, outliers can distort the distance function.

As shown in Table 4, we conducted another experiment in which we used only the *MIJA* features in the proposed approach, and obtained an average accuracy rate of 84.6%. We also evaluated the approach using both features, *MIRM* and *MIJA*, and obtained an average accuracy rate of 91.2%, showing that the combination of the two types of feature is much better than that of using only one feature. This is because, although subjects can change their speed of motion or orientation, the structure of the relative position of essential body parts usually remains the same during most of complicated actions. Note that, the effect of using only the *MIRM* features cannot be estimated since based on our algorithm in Fig. 5, some simple actions (e.g., *horizontal arm wave*, *hammer*, and *draw x*) in the dataset are not associated to this type of feature.

To compare the classification accuracy of our approach with the technique in [53], we used their approach of partitioning the MSR Action3D dataset, where samples of Subjects (1, 2, 3, 4, 5) were used for training and the remaining samples associated with Subjects (6, 7, 8, 9, 10) for test data. The result obtained from our approach and the technique in [53] are comparable. More specifically, our approach could successfully classify 266 out of 291 test samples resulting in the accuracy of 91.4%. This makes the difference in the correct classification rate between [53] and our approach to be just one sample. However, our approach in this paper can handle very noisy data as shown in the last experiment in this section. We were unable to investigate this capability for the approach in [53] as it is not described in that paper.

Further experiments were carried out to verify the accuracy of the proposed approach. Different subjects used to make up the

training and the unseen test dataset can affect the classification accuracy of a specific algorithm. To remove this bias, we evaluated our approach using all possible combinations of five subjects out of ten ($^{10}C_5 = 252$) as different sets of training and the unseen test datasets respectively. Each of these combinations of training and unseen test dataset were used in the evaluation of our approach and the average classification accuracy from 252 runs of the algorithm using each of these combinations was 89.4%. This evaluation is different from those approaches (as listed in Table 4 and paper [53]) where one combination of subjects respectively in the training and unseen test datasets were used in their evaluation and demonstrated the robustness of our approach to variations in the training dataset.

Table 4
Comparison between the proposed method (in bold) and previous depth-based and skeleton-based approaches on the MSR Action3D dataset.

Method	Accuracy	Type
Action Graph on Bag of 3D Points [47]	74.7	Depth-based
Space–Time Occupancy Patterns [48]	84.8	Depth-based
Random Occupancy Patterns [49]	86.5	Depth-based
Oreifej and Liu [50]	88.8	Depth-based
Spatio-Temporal Depth Cuboid [18]	89.3	Depth-based
Super Normal Vector [20]	93.0	Depth-based
Ellis [52]	65.7	Skeleton-based
HOJ3D [33]	79.0	Skeleton-based
EigenJoints [27]	82.3	Skeleton-based
Dynamic Temporal Warping [36]	83.7	Skeleton-based
Action-let Ensemble [31]	88.2	Skeleton-based
Pose-based Action Recognition [51]	90.0	Skeleton-based
LCSS + MIJA	**84.6**	Skeleton-based
LCSS + MIJA + MIRM (the proposed method)	**91.2**	Skeleton-based

Table 5
Comparisons between characteristics of the proposed approach with those of Action-let Ensemble model [31].

	Action-let Ensemble model	The proposed approach
Features	Relative joint positions + LOP (skeleton + depth)	MIRM + MIJA (both skeleton)
Algorithm for discriminative features	Mining Action-let Ensemble	One algorithm for each feature type
Temporal structure of actions	Through using Fourier temporal pyramid	Captured directly via using the two types of feature
Classification	SVM	LCSS
Accuracy	88.2	91.2

Table 6
Comparisons between the proposed method (in bold) and previous approaches on the HDM05 dataset.

Method	Accuracy
HMIJ + SVM [38]	84.40
HMIJ + Nearest neighbour [38]	80.73
SMIJ + Nearest neighbour [38]	81.65
SMIJ + SVM [38]	82.57
Dynamic Temporal Warping [36]	82.08
The proposed method	**85.23**

To further investigate the performance of the proposed approach, we compared classification results from this study with those of previous approaches on the HDM05 mocap dataset. Authors in [38] proposed two action representation techniques and evaluated their performance on the dataset using SVM and nearest neighbour classifiers. As shown in Table 6, the highest accuracy of these two techniques when applied to the HDM05 dataset were estimated to be 84.40 and 82.57 for using SVM and nearest neighbour classifiers, respectively. During our experiments, we observed that for different action classes, different numbers of joints are involved. However, a fixed number of joints were considered in the techniques in [38] to calculate the body-joint features for different action classes. Consequently, features of joints that may not be active in some gesture classes are incorporated for action classification, hence the lower accuracy estimates compared to that of our proposed method. We also applied the method proposed in [36] against the dataset. As shown in Table 6, it had a lower accuracy than that of our proposed method because, as mentioned above, its distance estimates can be distorted by outliers.

In the last experiment, we compared the robustness of our classifier to that of the other most similar template matching technique used in the literature for skeletal based action recognition i.e., DTW. Zero-mean and statistically independent noise was added to the 3D joint positions of test samples in the MSR Action3D dataset. More precisely, we evaluated the influence of two types of noise, namely "*Salt and Pepper*" and "*Gaussian*" for 10 iterations to take variability of noise into account. As can be seen in Fig. 11(a) and (b), although the amount of noise increases and, consequently, the average classification rate of DTW method decreases, the proposed method is more robust to the increasing level of noise. Also, the DTW technique seems to be much more sensitive to '*Salt and Pepper*' noise as it shows substantial loss of accuracy in comparison with LCSS.

5. Conclusion

This paper has introduced an approach for action classification using skeletal tracking data and evaluated the approach using data captured from the Microsoft Kinect and mocap systems. After a

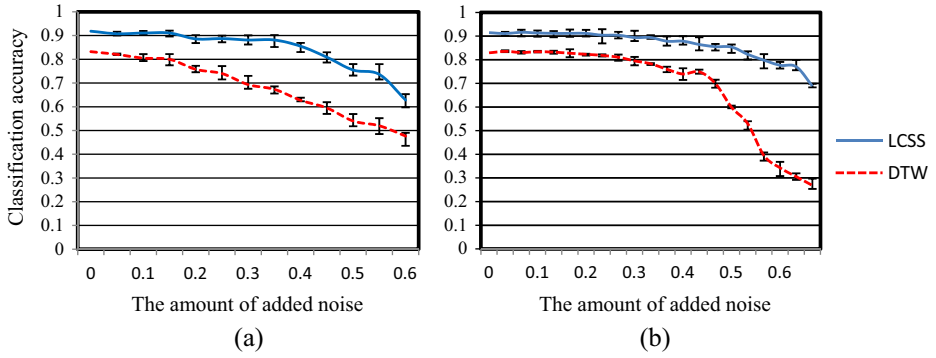

Fig. 11. Classification rates of the proposed approach and DTW after adding (a) Gaussian and (b) salt and paper noise to the 3D joint positions in the MSR Action3D dataset.

data representation step, the proposed approach learns the set of participating joints and joint pairs with discriminative trajectories and relative motions in each action class, respectively. Evaluation results have verified improved performance of the proposed method over some of the state-of-the-art approaches for action classification. Additionally, by using an extended formulation of LCSS as a similarity function, the proposed method is more robust to noise than previous techniques, as it efficiently allows outliers to be unmatched in the trajectories with different lengths. Our future work will include incorporating additional features for understanding more complex activities, such as those involving interactions with the environment.

References

[1] P. Antonakaki, D. Kosmopoulos, S.J. Perantonis, Detecting abnormal human behaviour using multiple cameras, Signal Process. 89 (9) (2009) 1723–1738.
[2] Y. Chen et al., Abnormal behavior detection by multi-SVM-based Bayesian network, in: International Conference on Information Acquisition, 2007, ICIA'07, IEEE, 2007.
[3] J.J. Corso et al., A practical paradigm and platform for video-based human–computer interaction, Computer 41 (5) (2008) 48–55.
[4] H. Seki, Fuzzy inference based non-daily behavior pattern detection for elderly people monitoring system, in: Conf. Proc. IEEE Eng. Med. Biol. Soc., 2009, 2009, pp. 6187–6192.
[5] Kinect Camera, 2013. <http://www.xbox.com/en-AU/Kinect>. <http://www.xbox.com/en-AU/Kinect> (cited 13.12.13).
[6] K. Lai et al., Sparse distance learning for object recognition combining RGB and depth information, in: IEEE International Conference on Robotics and Automation (ICRA), 2011, IEEE, 2011.
[7] Kinect for Windows SDK, 2013. <http://msdn.microsoft.com/en-us/library/hh855347.aspx> (cited 13.12.13).
[8] OpenNI, 2013. <http://www.openni.org/> (cited 13.12.13).
[9] J. Aggarwal, L. Xia, Human activity recognition from 3D data: a review, Pattern Recogn. Lett. (2014).
[10] M. Vlachos, G. Kollios, D. Gunopulos, Discovering similar multidimensional trajectories, in: Proceedings, 18th International Conference on Data Engineering, 2002, IEEE, 2002.
[11] S. Almotairi, E. Ribeiro, Human action recognition using temporal sequence alignment, in: International Conference on Computational Science and Computational Intelligence (CSCI), 2014, IEEE, 2014.
[12] M. Vrigkas, Action recognition by matching clustered trajectories of motion vectors, in: VISAPP (1), 2013.
[13] S.-Y. Jin, H.-J. Choi, Essential body-joint and atomic action detection for human activity recognition using longest common subsequence algorithm, in: Computer Vision-ACCV 2012 Workshops, Springer, 2013.
[14] I. Laptev, On space–time interest points, Int. J. Comput. Vis. 64 (2–3) (2005) 107–123.
[15] P.-C. Chung, C.-D. Liu, A daily behavior enabled hidden Markov model for human behavior understanding, Pattern Recogn. 41 (5) (2008) 1572–1580.
[16] W.-L. Lu, J.J. Little, Simultaneous tracking and action recognition using the PCA-HOG descriptor, in: The 3rd Canadian Conference on Computer and Robot Vision, 2006, IEEE, 2006.
[17] B. Ni, G. Wang, P. Moulin, RGBD-HuDaAct: a color-depth video database for human daily activity recognition, in: Consumer Depth Cameras for Computer Vision, Springer, 2013, pp. 193–208.
[18] L. Xia, J. Aggarwal, Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013, IEEE, 2013.
[19] X. Yang, C. Zhang, Y. Tian, Recognizing actions using depth motion maps-based histograms of oriented gradients, in: Proceedings of the 20th ACM International Conference on Multimedia, 2012, p. ACM.
[20] X. Yang, Y. Tian, Super normal vector for activity recognition using depth sequences, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, 2014.
[21] S. Hadfield, R. Bowden, Hollywood 3D: recognizing actions in 3D natural scenes, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013, IEEE, 2013.
[22] G. Willems, T. Tuytelaars, L. Van Gool, An efficient dense and scale-invariant spatio-temporal interest point detector, in: Computer Vision–ECCV 2008, Springer, 2008, pp. 650–663.
[23] O. Oshin, A. Gilbert, R. Bowden, Capturing the relative distribution of features for action recognition, in: IEEE International Conference on Automatic Face & Gesture Recognition and Workshops (FG 2011), 2011, IEEE, 2011.
[24] G. Johansson, Visual perception of biological motion and a model for its analysis, Percept. Psychophys. 14 (2) (1973) 201–211.
[25] K. Lai, J. Konrad, P. Ishwar, A gesture-driven computer interface using Kinect, in: IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI), 2012, IEEE, 2012.
[26] T. Arici et al., Robust gesture recognition using feature pre-processing and weighted dynamic time warping, Multimed. Tools Appl. (2013) 1–18.
[27] X. Yang, Y. Tian, Effective 3D action recognition using eigenjoints, J. Vis. Commun. Image Represent. 25 (1) (2014) 2–11.
[28] V. Bloom, D. Makris, V. Argyriou, G3D: a gaming action dataset and real time action recognition evaluation framework, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2012, IEEE, 2012.
[29] C. Zhang, Y. Tian, RGB-D camera-based daily living activity recognition, J. Comput. Vis. Image Process. 2 (4) (2012) 12.
[30] S.Z. Masood et al., Measuring and reducing observational latency when recognizing actions, in: IEEE International Conference on Computer Vision Workshops (ICCV Workshops), 2011, IEEE, 2011.
[31] J. Wang et al., Mining actionlet ensemble for action recognition with depth cameras, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012, IEEE, 2012.
[32] J. Sung et al., Unstructured human activity detection from RGBD images, in: IEEE International Conference on Robotics and Automation (ICRA), 2012, IEEE, 2012.
[33] L. Xia, C.-C. Chen, J. Aggarwal, View invariant human action recognition using histograms of 3D joints, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2012, IEEE, 2012.
[34] L. Han et al., Discriminative human action recognition in the learned hierarchical manifold space, Image Vis. Comput. 28 (5) (2010) 836–849.
[35] S. Sempena, N.U. Maulidevi, P.R. Aryan, Human action recognition using dynamic time warping, in: International Conference on Electrical Engineering and Informatics (ICEEI), 2011, 2011, p. IEEE.
[36] M. Reyes, G. Dominguez, S. Escalera, Featureweighting in dynamic timewarping for gesture recognition in depth data, in: IEEE International Conference on Computer Vision Workshops (ICCV Workshops), 2011, IEEE, 2011.
[37] F. Lv, R. Nevatia, Recognition and segmentation of 3-D human action using HMM and multi-class adaboost, in: Computer Vision–ECCV 2006, Springer, 2006, pp. 359–372.
[38] F. Ofli et al., Sequence of the most informative joints (SMIJ): a new representation for human skeletal action recognition, J. Vis. Commun. Image Represent. 25 (1) (2014) 24–38.
[39] M. Pierobon et al., Clustering of human actions using invariant body shape descriptor and dynamic time warping, in: IEEE Conference on Advanced Video and Signal Based Surveillance, 2005, AVSS 2005, IEEE, 2005.
[40] J. Wang, H. Zheng, View-robust action recognition based on temporal self-similarities and dynamic time warping, in: IEEE International Conference on Computer Science and Automation Engineering (CSAE), 2012, IEEE, 2012.
[41] A. Corradini, Dynamic time warping for off-line recognition of a small gesture vocabulary, in: Proceedings, IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems 2001, IEEE, 2001.
[42] L.A. Shalabi, Z. Shaaban, B. Kasasbeh, Data mining: a preprocessing engine, J. Comput. Sci. 2 (9) (2006) 735.
[43] M. Raptis, D. Kirovski, H. Hoppe, Real-time classification of dance gestures from skeleton animation, in: Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, ACM, 2011.
[44] S. Monir, S. Rubya, H.S. Ferdous, Rotation and scale invariant posture recognition using Microsoft Kinect skeletal tracking feature, in: 12th International Conference on Intelligent Systems Design and Applications (ISDA), 2012, IEEE, 2012.
[45] C.E. Shannon, A mathematical theory of communication, ACM SIGMOBILE Mobile Computing and Communications Review 5 (1) (2001) 3–55.
[46] M. Müller, et al., Documentation Mocap Database HDM05, 2007.
[47] W. Li, Z. Zhang, Z. Liu, Action recognition based on a bag of 3D points, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2010, IEEE, 2010.
[48] A.W. Vieira et al., Stop: space–time occupancy patterns for 3D action recognition from depth map sequences, in: Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, Springer, 2012, pp. 252–259.
[49] J. Wang et al., Robust 3d action recognition with random occupancy patterns, in: Computer Vision–ECCV 2012, Springer, 2012, pp. 872–885.
[50] O. Oreifej, Z. Liu, HON4D: histogram of oriented 4D normals for activity recognition from depth sequences, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013, IEEE, 2013.
[51] C. Wang, Y. Wang, A.L. Yuille, An approach to pose-based action recognition, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013, 2013, p. IEEE.
[52] C. Ellis et al., Exploring the trade-off between accuracy and observational latency in action recognition, Int. J. Comput. Vis. 101 (3) (2013) 420–436.
[53] M. Zanfir, M. Leordeanu, C. Sminchisescu, The moving pose: an efficient 3D kinematics descriptor for low-latency action recognition and detection, in: IEEE International Conference on Computer Vision (ICCV), 2013, 2013.