Hindawi
Journal of Robotics
Volume 2017, Article ID 7610417, 8 pages
https://doi.org/10.1155/2017/7610417

Research Article

Tracking a Subset of Skeleton Joints: An Effective Approach towards Complex Human Activity Recognition

Muhammad Latif Anjum,[1] Stefano Rosa,[2] and Basilio Bona[2]

[1]*School of Electrical Engineering and Computer Science (SEECS), National University of Sciences and Technology (NUST), H-12, Islamabad, Pakistan*
[2]*Department of Control and Computer Engineering (DAUIN), Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy*

Correspondence should be addressed to Muhammad Latif Anjum; latif.anjum@seecs.edu.pk

Received 6 September 2016; Revised 18 November 2016; Accepted 12 December 2016; Published 17 January 2017

Academic Editor: Yinlai Jiang

We present a robust algorithm for complex human activity recognition for natural human-robot interaction. The algorithm is based on tracking the position of selected joints in human skeleton. For any given activity, only a few skeleton joints are involved in performing the activity, so a subset of joints contributing the most towards the activity is selected. Our approach of tracking a subset of skeleton joints (instead of tracking the whole skeleton) is computationally efficient and provides better recognition accuracy. We have developed both manual and automatic approaches for the selection of these joints. The position of the selected joints is tracked for the duration of the activity and is used to construct feature vectors for each activity. Once the feature vectors have been constructed, we use a Support Vector Machines (SVM) multiclass classifier for training and testing the algorithm. The algorithm has been tested on a purposely built dataset of depth videos recorded using Kinect camera. The dataset consists of 250 videos of 10 different activities being performed by different users. Experimental results show classification accuracy of 83% when tracking all skeleton joints, 95% when using manual selection of subset joints, and 89% when using automatic selection of subset joints.

1. Introduction

After a successful revolution in industrial robotics, the focus of the robotics community has now been shifted to humanoids and social robotics. For robots to invade social sphere of humans, the most important task for them is to understand what is going on around them or more specifically to understand what activities the humans around them are performing. A successful recognition of the activities being performed by humans will enable the robots to respond appropriately and is, therefore, an important step towards natural human robot interaction. This paper makes an attempt to contribute to the field and presents an algorithm to help robots classify activities being performed in front of them. The key advantage of our approach to activity recognition based on skeleton tracking is the selection of joints most relevant to the activity being performed. This not only reduces the computational complexity of the algorithm but also enhances the recognition accuracy as shown in experimental results. In the previous work ([1, 2]), we presented a method for skeleton tracking based activity classification based on the manual selection of subset joints depending upon the activity. With this paper, we extend that approach to automatic selection of subset joints most relevant to the activity. We also improve our dataset to include more activities and tested our algorithm on an activity involving two individuals. Our set of activities include *waving, checking wrist watch, doing a sit-stand exercise, sitting in a chair and drinking water, picking something from ground, pointing in a direction*, and *handshake between two individuals*. Additionally, we test our algorithm on three signals from umpire in a Cricket match.

We use the Robot Operating System (ROS) (http://www.ros.org/) to develop our algorithm. We use a modified version of the OpenNI skeleton tracker (http://wiki.ros.org/openni_tracker) to track the position of selected joints. As opposed to

other approaches [3], we are not using the position of all joints to construct feature vectors. Since only a few joints are undergoing a change in position for any given activity (considered as the joints contributing towards the activity), we select only those joints contributing the most towards an activity to construct feature vectors, thereby increasing computational efficiency and recognition performance of the algorithm. We present manual as well as automatic approaches for the selection of these joints and the experimental results for both approaches have been separately presented. The experimental results have been obtained on a purposely built dataset of activities. The dataset has been made public and constructed specifically to suit the requirements of skeleton tracking using ROS based OpenNI skeleton tracker. The contributions of this paper are (1) presenting a computationally efficient activity recognition algorithm based on tracking the position of a subset of skeleton joints, (2) provision of ROS based OpenNI skeleton tracker friendly RGB-D dataset of human activities, and (3) classification of activities involving multiple individuals.

The dataset along with all the ROS packages have been made public and can be accessed through the website of our research group, LabRob (http://www.polito.it/Labrob). The dataset contains video files of all activities in OpenNI file format (.oni).

2. Related Work

The research in activity recognition took a great step forward after the advent of cameras that can capture depth images and videos [4]. Microsoft Kinect is a low cost and readily available sensor and is being used extensively in many computer vision and robotics applications [5]. Zhu and Fujimura demonstrated the advantage of using depth images for human body pose tracking [6]. Their approach used key-point method where they detected and tracked anatomical landmarks (key-points) of human body to reconstruct the human pose. Activity recognition using depth videos has also been presented by Koppula et al. [7]. They have combined skeleton tracking and object affordances to construct feature vectors and have then used structural support vector machines (SSVM) for training and testing. Their results were obtained after testing the algorithm on a purposely built dataset of their own which included only the activities involving one individual. Droeschel et al. presented a person awareness and gesture recognition approach for joint attention in a domestic environment [8]. They used time-of-flight cameras to classify between *showing* and *pointing* gestures and presented their results with high accuracy.

The fact that all skeleton joints do not contribute equally towards an activity was used by Jiang et al. [9]. They analyzed mean contribution of each skeleton joint for various action classes and utilized the contribution ratio of the joints to classify the actions. Their approach is efficient as shown by the experimental results but they are still tracking the positions of all skeleton joints. The idea of using selected joints for activity recognition has recently been utilized by Wu et al. [10]. They only focus on informative body parts such as head and hands to construct Histograms of Located Displacements (HOLD) and Local Depth Motion Maps (L-DMM) based Gabor representation. The discriminative nature of Gabor representations help classify human activities. Our approach is an extension of the idea of using selected skeleton joints for activity recognition.

Figure 1: A user makes a surrender pose in front of a camera.

The remainder of this paper is organized as follows. Section 3 describes method for tracking the skeleton joints, followed by Section 4, where the construction scheme for feature vectors is discussed. Both manual and automatic approaches to the selection of subset joints are discussed in this section. Section 5 describes the dataset we created to experimentally test our algorithm along with other available RGB-D datasets. The SVM training and testing procedure is described in Section 6, followed by the description of experimental results in Section 7. Section 8 concludes the paper.

3. Tracking Joints' Positions

OpenNI ROS package for skeleton tracking is available to all ROS users. The package works with a Kinect camera connected to a PC and can track fifteen joints and body parts in human skeleton. These joints include both hands, both elbows, head, neck, both shoulders, torso, left and right hips, both knees, and both feet. The 3D positions and rotation quaternions of all these joints in space are published with reference to the center of camera frame, */openni_depth_frame*, as a set of ROS transforms (*/tf*). These joints are shown in Figure 1 where an individual makes a surrender pose in front of the camera.

We have modified the OpenNI skeleton tracker package to work with off-line recorded videos in our dataset. The videos have been recorded in OpenNI file format with the extension (.oni) and contain both the RGB and depth information needed for OpenNI skeleton tracker.

4. Construction of Feature Vectors for SVM

In order to use SVM, we need to extract certain features from activity videos and construct feature vectors corresponding to every activity video in our dataset. We use the position of selected joints in every activity as features. Using position of all joints to construct feature vectors is inefficient because,

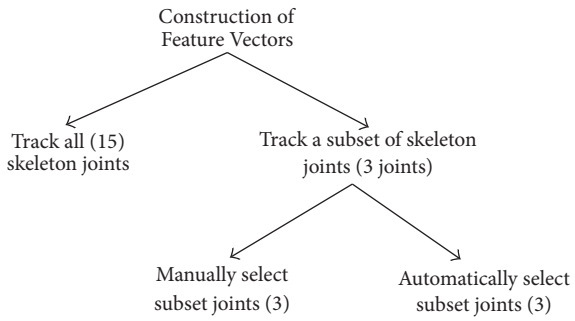

Figure 2: Methods developed for selecting feature vectors.

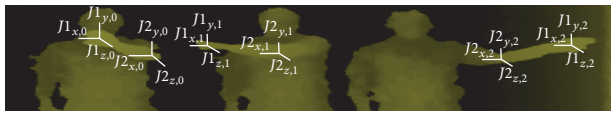

Figure 3: Position of two joints across three frames.

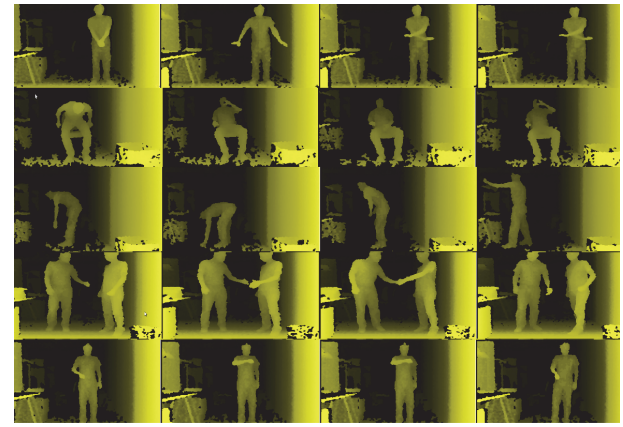

Figure 4: Screenshot images at four different positions during the dead ball, sit and drink, pick something from ground, handshake, and check watch activities,respectively.

for any given activity, there is only a subset of joints contributing towards the activity. For example, for *waving hello* activity, only the position of right hand and right elbow is changing while the position of all other joints remains the same. Using the position of all joints to construct feature vectors will increase the similarity index of feature vectors that will decrease recognition efficiency. We have, however, experimented with both approaches and tabulated results for both. Figure 2 shows possible methods to select feature vectors. For the selection of subset joints, we have developed both manual and automatic approaches explained in Sections 3 and 4.

4.1. Tracking All Skeleton Joints. This approach is the simplest where we use the position of all fifteen (15) joints with reference to *torso* to construct feature vectors for each activity. Let these joints be $J1, J2, J3, \ldots, J15$. We have the 3D position of each joint in successive frames during the whole activity (shown in Figure 3 with three successive frames). If we represent the x component of the position of joint $J1$ in frame number 0 as $J1_{x,0}$, y and z components as $J1_{y,0}$ and $J1_{z,0}$, respectively, we can formulate feature vectors (FV) as in (1). The subscripts $\{0, 1, \ldots, n\}$ are the frame number

$$\begin{aligned}\mathrm{FV} = \{&A1, \{(J1_{x,0}, J2_{x,0}, \ldots, J15_{x,0}),\\ &(J1_{y,0}, J2_{y,0}, \ldots, J15_{y,0}), (J1_{z,0}, J2_{z,0}, \ldots, J15_{z,0})\},\\ &\{(J1_{x,1}, J2_{x,1}, \ldots, J15_{x,1}), (J1_{y,1}, J2_{y,1}, \ldots, J15_{y,1}),\\ &(J1_{z,1}, J2_{z,1}, \ldots, J15_{z,1})\}, \ldots,\\ &\{(J1_{x,n}, J2_{x,n}, \ldots, J15_{x,n}), (J1_{y,n}, J2_{y,n}, \ldots, J15_{y,n}),\\ &(J1_{z,n}, J2_{z,n}, \ldots, J15_{z,n})\}\},\end{aligned} \tag{1}$$

where n is the number of frames in a given activity and its value depends on the length of each activity. The first element, $A1$, of each feature vector is the label of the activity required for SVM based training program. Each feature vector FV is, therefore, a row vector and has $(45 \times n) + 1$ elements if we use all skeleton joints as given in (1).

A word is in order regarding the selection of *torso* as the reference point. If we consider camera position as the reference, joints' position will depend on the position of user in front of the camera. This will result in different recognition results for the same activity being performed at different positions in front of the camera. To avoid this, reference point is selected on the user's body to ensure that the position of joints is not dependent on position of the user in front of the camera; that is, the user can stand near the camera, far from it, or walk in front of the camera and the algorithm works in all the cases as long as the whole skeleton remains visible to the camera. *Torso* provides a good reference point because it is located in the middle of human skeleton and remains stationary during most of common human activities. It does change its position during our *sit-stand* activity but experimental results show it works well even for activities involving motion of torso.

4.2. Tracking a Subset of Skeleton Joints. As has been discussed in Section 4.1, using all skeleton joints to construct feature vectors adds to the similarity index of feature vectors resulting in poor efficiency. Our approach is to select only those joints contributing significantly towards the activity to construct feature vectors. We have to decide how many joints are required to constitute the subset for feature vector construction. Repeated experiments with all activities have shown that three joints are sufficient for the construction of feature vectors for the activities in our dataset. Selecting the number of joints could be tricky: tracking fewer joints involves the risk of losing important information while tracking more joints (specifically the ones not undergoing any motion) involves the risk of increasing similarity index between activities. Three joints in each activity have been selected to form the subset of joints for feature vector construction. Once the joints in the subset have been selected,

a modified version of (1) is used to construct feature vectors as shown in

$$\begin{aligned}\mathrm{FV} = \{A1, \{(J1_{x,0}, J2_{x,0}, J3_{x,0}), (J1_{y,0}, J2_{y,0}, J3_{y,0}),\\ (J1_{z,0}, J2_{z,0}, J3_{z,0})\}, \{(J1_{x,1}, J2_{x,1}, J3_{x,1}),\\ (J1_{y,1}, J2_{y,1}, J3_{y,1}), (J1_{z,1}, J2_{z,1}, J3_{z,1})\}, \ldots,\\ \{(J1_{x,n}, J2_{x,n}, J3_{x,n}), (J1_{y,n}, J2_{y,n}, J3_{y,n}),\\ (J1_{z,n}, J2_{z,n}, J3_{z,n})\}\}.\end{aligned} \tag{2}$$

Since we have reduced the number of joints to only three, each feature vector FV here will be a row vector with $(9 \times n) + 1$ elements as given in (2). The number of elements in each feature vector has been reduced five times with our approach of tracking a subset of skeleton joints, thereby reducing the computational cost. For the selection of subset joints, we have developed both manual and automatic selection approaches described below.

4.2.1. Manual Selection of Subset Joints. A manual approach for selecting a subset of skeleton joints has been developed to test the accuracy of our algorithm. Only those joints that are contributing the most towards the activity are selected and the selection is made based on human intuition and the knowledge of the activity. For example, *waving* and *checking wrist watch* activities include a distinct motion of right arm, so the position of right hand and right elbow can be tracked for these two activities. Similarly, the position of right hand and right knees can be tracked for a *leg bye signal* activity. Since we have an activity that involves two users *(handshake)*, we have tracked the position of right hand and right elbow of the second user in addition to the right hand of first user for a *handshake* activity. Some complex and long activities like *sitting on a chair and drinking* and *sit-stand* involve the motion of many joints while other activities like *waving hello* and *checking wrist watch* involve motion of only two joints. Table 1 lists the joints tracked during manual selection for each activity along with the point of reference.

Table 1: The summary of the joints and their reference points used during manual selection of subset joints where the /right_hand_2 shows the right hand of second user during handshake activity.

Activity	Joints tracked	Reference
Pointing in a direction	/left_hand	/torso
	/left_elbow	/torso
	/left_hand	/torso
Sitting on a chair and drinking from bottle	/right_elbow	/torso
	/right_hand	/torso
	/left_hand	/torso
Picking an object from ground	/right_hand	/torso
	/head	/torso
	/right_shoulder	/torso
Sit-stand exercise	/right_foot	/torso
	/left_foot	/torso
	/left_knee	/torso
Checking wrist watch	/left_hand	/torso
	/left_elbow	/torso
	/left_hand	/torso
Waving hello	/right_hand	/torso
	/right_elbow	/torso
	/right_elbow	/torso
Handshake between two individuals	/right_hand	/torso
	/right_hand_2	/torso_2
	/right_elbow	/torso
Dead ball signal	/right_hand	/torso
	/left_hand	/torso
	/head	/torso
Four signal	/right_hand	/torso
	/right_elbow	/torso
	/right_elbow	/torso
Leg bye signal	/right_hand	/torso
	/right_knee	/torso
	/right_knee	/torso

4.2.2. Automatic Selection of Subset Joints. Robots do not have intuition and prior knowledge of the activity and therefore the manual joint selection strategy cannot be used in practice. We need a system to automatically select joints from the skeleton contributing the most towards any given activity. For this purpose, we track the position of all joints with respect to a single reference, (*/torso*) and constitute position vectors, ν_i, of each joint as given in

$$\begin{aligned}\nu_i = \{(Ji_{x,0}, Ji_{y,0}, Ji_{z,0}), (Ji_{x,1}, Ji_{y,1}, Ji_{z,1}), \ldots,\\ (Ji_{x,n}, Ji_{y,n}, Ji_{z,n})\},\end{aligned} \tag{3}$$

where ν_i is the position vector of ith joint.

Once the position vectors of all joints have been obtained with respect to one single reference, we compute the variance of all position vectors in the activity. The joints with highest variance of position vectors are the ones undergoing highest motion (change in position) and are therefore contributing the most towards the activity. Three joints with highest variances are considered for the construction of feature vectors. Once joints are selected, (2) (only for three selected joints) is used to construct feature vectors.

Some activities such as *sit-stand* activity may involve motion of some joints that do not define that activity. For example one person performing *sit-stand* activity might be moving his/her hands along the way, while another might not. The movement of hands therefore does not define the *sit-stand* activity. Our automatic joint selection procedure, however, selects three joints undergoing highest motion irrespective of the nature of joints. Therefore, joints selected could be different for the same activity being performed by different individuals.

Figure 5: Screenshot images at four different positions during the four signal, leg bye signal, sit stand, wave, and point in a direction activities, respectively.

5. Creating the Dataset

Since most of the work in gesture and activity recognition has been previously done using RGB videos and images, there are only a few activity datasets available involving RGB-D videos. An RGB-D dataset, CAD-120, was presented by Koppula et al. [7]. Their dataset is provided in two formats: as correlated RGB and depth images and as RGB-D text format. Both of these formats require further processing to convert the correlated RGB and depth images into an OpenNI video format to run ROS OpenNI skeleton tracker. Another such dataset, presented by [11], is not suitable for OpenNI skeleton tracker because their videos do not start with the surrender pose (Figure 1). The surrender pose is necessary for ROS based OpenNI tracker to calibrate the user and start tracking.

We have therefore constructed a dataset of our own and made it public through the website of our group, http://www.polito.it/Labrob. Our dataset is purposely built to be used with ROS based OpenNI skeleton tracker and should serve as a reference for future research in Kinect based activity recognition. The dataset was initially presented in [1] and we have now added few more activities to the dataset. The dataset now consists of a total of 10 activities being performed by different users. The activities include 6 daily life activities: *waving hello, checking wrist watch, picking something from ground and placing it over a cupboard, sit-stand, sitting on a chair and drinking from bottle*, and *pointing in a direction*. We have additionally included a *handshake between the two individuals* activity to test our algorithm for activities involving multiple individuals. Finally our dataset includes three distinct signals from the umpire in a Cricket match. These signals include *four signal, leg bye signal,* and *dead ball signal*. Screenshot images from these activities are shown in Figures 4 and 5. We have recorded 25 videos of each activity in the dataset so the dataset contains a total of 250 videos. The videos are available in OpenNI depth video format (.oni) with the resolution of 640×480.

Selection of a particular activity to be included in our dataset required a bit of discussion. There are countless daily life human activities which can be included in our dataset. Since our approach to activity classification is based on tracking position of skeleton joints, our first guiding principle was to select activities that involve movement of skeleton joints. The selection of signals from umpire in a Cricket match was influenced by the fact that those signals involved extensive movement of joints (both hands and legs). There is no end of the possible human daily life activities, but 10 activities were considered sufficient enough to verify our approach. Other activities can be added to the dataset in future.

6. Activity Classification Using SVM

Once we have constructed the feature vectors of each activity using (1) or (2) (whichever is applicable), we use Support Vector Machines (SVM) to classify the activities. The features data set needs to be split into training and testing data for SVM. Out of 25 videos of each activity, we use 15 videos of each activity (a total of 150 videos) for training the SVM and the algorithm is tested on 10 remaining videos of each activity (a total of 100 videos). The SVM training data matrix is constructed using

$$\text{trainData} = \begin{bmatrix} (\text{FV1}_1, \text{FV1}_2, \text{FV1}_3, \ldots, \text{FV1}_{15})^T \\ (\text{FV2}_1, \text{FV2}_2, \text{FV2}_3, \ldots, \text{FV2}_{15})^T \\ \vdots \\ (\text{FV10}_1, \text{FV10}_2, \text{FV10}_3, \ldots, \text{FV10}_{15})^T \end{bmatrix}, \quad (4)$$

where $\text{FV1}, \text{FV2}, \ldots, \text{FV10}$ represent the feature vectors of corresponding activities constructed individually using (1) or (2), while the subscript represents the number of video of each activity. Since we have used first 15 videos in our dataset for training, the subscript goes from 1 to 15. As a result, the training data matrix will have an order of $150 \times ((9 \times n) + 1)$, if (2) is used, or an order of $150 \times ((45 \times n) + 1)$, if (1) is used. Since n is the number of frames in a given activity and its value depends upon the length of each activity, it will be different for each activity resulting in various rows of training data matrix (4) having different number of elements. To counter this, the number of columns in training data matrix is based on maximum value of n (for the longest activity) and remaining matrices are padded with zeros. The testing data matrix for SVM is similarly constituted using

$$\text{testData} = \begin{bmatrix} (\text{FV1}_{16}, \text{FV1}_{17}, \text{FV1}_{18}, \ldots, \text{FV1}_{25})^T \\ (\text{FV2}_{16}, \text{FV2}_{17}, \text{FV2}_{18}, \ldots, \text{FV2}_{25})^T \\ \vdots \\ (\text{FV10}_{16}, \text{FV10}_{17}, \text{FV10}_{18}, \ldots, \text{FV10}_{25})^T \end{bmatrix}, \quad (5)$$

where the testing data matrix is of the order of $100 \times ((9 \times n) + 1)$, if (2) is used, or of the order of $100 \times ((45 \times n) + 1)$ if (1)

is used. This matrix will classify all 100 test videos in one go. The number of rows of test data matrix can be changed based on the number of videos to be classified. Clearly, none of the video used in training is used for testing the algorithm.

Support vector machines are machine learning tools that are used to classify data consisting of high dimensional feature space. The theory is based on selecting optimal hyperplanes that separate various classes of data. In general, the hyperplanes can be represented by equation $f(x) = \beta_0 + \beta^T x$, where β and β_0 are weight vectors and bias, respectively, and x consists of training data points closest to the hyperplane. The SVM algorithm selects β_0 and β to maximize the distance between hyperplanes and x, called the margin. The margin can be expressed as $M = 2/\|\beta\|$ where $\|\beta\|$ is the norm of β. Hence the problem of maximizing M is essentially equivalent to problem of minimizing a β dependent function $F(\beta)$ subject to some constraints. This can be expressed by optimization equation:

$$\begin{aligned} &\underset{\beta,\beta_0}{\text{minimize}} \quad F(\beta) = \frac{2}{\|\beta\|^2} \\ &\text{subject to} \quad A_i\left(\beta_0 + \beta^T x_i\right) \geq 1 \quad \forall i, \end{aligned} \tag{6}$$

where A_i are the labels introduced in the training set. This optimization problem is solved using Lagrange multipliers to find the optimal values of β and β_0. For further readings on SVM, [12] can be consulted.

Since we have multiple categories for classification, we have used SVM multiclass classifier available in OpenCV library, LibSVM. Gaussian kernel has been used for training the algorithm. The algorithm has been implemented using ROS environment and a final ROS package for SVM based training and testing has been made available. The steps required to classify any incoming activity are summarized in flow diagram shown in Figure 6.

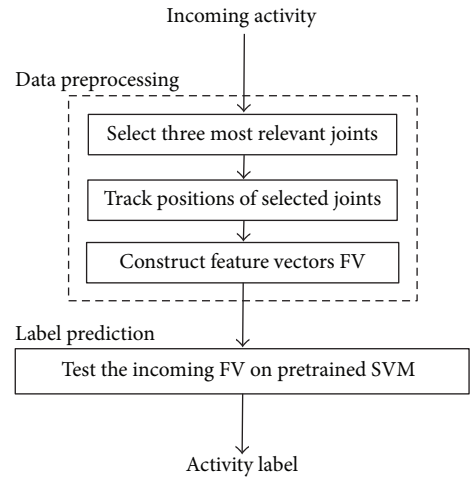

Figure 6: Flow diagram detailing the steps required to classify an incoming activity.

7. Experimental Results

Experimental results were obtained on a subset of activity videos that were not used for SVM based training. Experimental results are separately presented for three cases: (a) when tracking all skeleton joints, (b) when manually selecting a subset of skeleton joints to be tracked, and (c) when automatically selecting a subset of joints to be tracked.

7.1. Recognition Results When Using All Skeleton Joints. We selected 10 videos for each activity for testing the algorithm that were not previously used for training and therefore a total of 100 videos of 10 activities. The SVM based testing algorithm successfully classified 83 videos out of 100, resulting in an accuracy of 83%. As has been emphasized earlier, using all skeleton joints for feature vectors construction only adds to the similarity index of the feature vectors resulting in lower accuracy. Table 2 shows the confusion matrix for this case.

The experimental results indicate the problem with tracking all skeleton joints. The activities are being confused with each other, especially the activities which are similar or involve motion of same joints. This is because most of the joints being tracked are stationary in all activities and are adding to the similarity index of the activities resulting in poor recognition accuracy.

Table 2: Confusion matrix for activity recognition when using all skeleton joints.

No.	Activity	1	2	3	4	5	6	7	8	9	10
1	Point in a direction	8								2	
2	Pick from ground		7		2						1
3	Check watch			9			1				
4	Sit-stand				9	1					
5	Sit and drink				2	8					
6	Wave				1		8			1	
7	Handshake							9	1		
8	Dead ball signal		1						9		
9	Four signal						2			8	
10	Leg bye signal							1		1	8

7.2. Recognition Results When Using a Subset of Skeleton Joints. We have two strategies for selecting subset joints. Experimental results are presented separately for both of them.

7.2.1. Recognition Results for Manual Selection of Subset Joints. The same videos were used to test the algorithm for manual selection of subset joints. The SVM based testing algorithm successfully classified 95 videos out of 100, resulting in an accuracy of 95%. Table 3 shows the confusion matrix for this case. As has been explained in Section 4, the manual selection of joints is made based on the intuition and the knowledge of the activity, so only those joints having distinct motion pattern in any activity were selected resulting in a higher accuracy.

The activities that are remarkably different based on the motion of joints have been classified with 100% accuracy. These activities include *pointing in a direction*, *picking from*

Table 3: Confusion matrix for activity recognition when using manual selection of joints.

No.	Activity	1	2	3	4	5	6	7	8	9	10
1	Point in a direction	10									
2	Pick from ground		10								
3	Check watch			10							
4	Sit-stand				10						
5	Sit and drink				1	9					
6	Wave						9			1	
7	Handshake							10			
8	Dead ball signal		1						9		
9	Four signal						2			8	
10	Leg bye signal										10

Table 4: Confusion matrix for activity recognition when using automatic selection of joints.

No.	Activity	1	2	3	4	5	6	7	8	9	10
1	Point in a direction	10									
2	Pick from ground		10								
3	Check watch			8			2				
4	Sit-stand				10						
5	Sit and drink				1	9					
6	Wave						8		1	1	
7	Handshake			1				9			
8	Dead ball signal			2					8		
9	Four signal						1		1	8	
10	Leg bye signal									1	9

ground, *checking watch*, *sit stand*, *leg bye signal,* and *handshake*. Notable confusions are *waving hello* and *four signal* because both of these activities include the motion of right hand in a not so distinct trajectory.

7.2.2. Recognition Results for Automatic Selection of Subset Joints. Automatic selection of joints is required for the robots to use the system in real time. As has been explained in Section 4, all joints are tracked with respect to a single reference and three joints with highest variances in position vectors are selected. Experimental results show that our algorithm successfully classifies 89 out of 100 test videos resulting in an accuracy of 89%. Table 4 shows the confusion matrix when using automatic joint selection.

7.3. Results Discussion. The experimental results show that tracking a subset of skeleton joints easily outperforms the approach of tracking whole skeleton. The selection of these subset joints is however tricky and we have presented both manual and automatic approaches to this. The experimental results show that the algorithm is more accurate when using manual selection of joints as compared to automatic selection of joints. This is understandable given the fact that joints are manually selected to make their motion trajectories distinct as possible for each activity. The results obtained with automatic selection of joints are also very encouraging and we are confusing only with activities that are similar to each other based on the motion of the joints involved. The algorithm has not only been tested on simple and small gestures but also on fairly long and complex activities. For example, the *sit and drink* activity involves a person sitting down on a chair, picking up a bottle of water, and then drinking from it. The experimental results are very encouraging and show the potential of the use of skeleton tracking for activity recognition. The skeleton tracking using depth images and videos is strikingly simple that eliminates many conventional headaches one encounters while using RGB images and videos such as background scene, colors, and lighting conditions. The approach is very useful in restricted social environments where robots have to identify a fixed number of activities and gestures and respond accordingly. For example, a coffee serving robot in a coffee bar can be trained with a specific set of activities and gestures the customers usually perform.

The question of what will happen if an unknown activity (an activity on which the algorithm has not been previously trained) is given as input to the classifier needs a bit of discussion. The SVM, as any other machine learning algorithm, requires training on all possible outcomes. If an activity from outside the training set comes in, the classifier will still classify it to one of the activities it was trained with. The unknown activity will be labeled as the one it resembles the most from the training set. Our approach, as has been written already, is useful in a restricted environment where robots have to classify a fixed number of activities. One possible way out is to train the algorithm for the negative world that includes all activities not included in the training set. That will require an extensive dataset of negative world activities and extensive research.

8. Conclusion

All human gestures and activities somewhat involve the motion of some of their joints and body parts. Any approach to activity recognition based on skeleton tracking, therefore, has the potential to classify any human gesture or activity. We have presented our approach to activity recognition for natural human-robot interaction where we use a subset of human skeleton joints to differentiate between the activities being performed. This subset is constituted using the skeleton joints contributing the most towards an activity. The selection of these joints can be intuitively made by humans while training the robot. We have also presented an automatic approach to the selection of these subset joints during the training phase. Experimental results for both approaches have been presented. Once the subset of joints is constituted, we use the 3D position of these joints for the period of the activity to construct feature vectors which are then used for SVM based training and testing of the approach. We have been able to obtain an accuracy of 95% when using manual selection of joints and an accuracy of 89% when using automatic selection of joints. The recognition accuracy when tracking all skeleton joints is also evaluated for comparison and is found to be lower than our algorithm.

Disclosure

This research was conducted during the PhD study of one of the authors at Politecnico di Torino, Italy, and was funded by Higher Education Commission (HEC), Government of Pakistan through its Faculty Development program, HRD-UESTPs/UETs, 2012–2015.

Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] M. L. Anjum, O. Ahmad, S. Rosa, J. Yin, and B. Bona, "Skeleton tracking based complex human activity recognition using kinect camera," in *Proceedings of the 6th International Conference on Social Robotics (ICSR '14)*, pp. 23–33, Sydney, Australia, 2014.

[2] M. L. Anjum, *Vision tracking and activity recognition: towards natural human-robot interaction [Ph.D. thesis]*, Department of Mechatronics, Politecnico di Torino, Torino, Italy, 2015.

[3] L. A. Schwarz, A. Mkhitaryan, D. Mateus, and N. Navab, "Human skeleton tracking from depth data using geodesic distances and optical flow," *Image and Vision Computing*, vol. 30, no. 3, pp. 217–226, 2012.

[4] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "RGB-D mapping: using kinect-style depth cameras for dense 3D modeling of indoor environments," *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 647–663, 2012.

[5] R. A. El-Laithy, J. Huang, and M. Yeh, "Study on the use of Microsoft Kinect for robotics applications," in *Proceedings of the IEEE/ION Position, Location and Navigation Symposium (PLANS '12)*, pp. 1280–1288, Myrtle Beach, SC, USA, April 2012.

[6] Y. Zhu and K. Fujimura, "A bayesian framework for human body pose tracking from depth image sequences," *Sensors*, vol. 10, no. 5, pp. 5280–5293, 2010.

[7] H. S. Koppula, R. Gupta, and A. Saxena, "Learning human activities and object affordances from RGB-D videos," *International Journal of Robotics Research*, vol. 32, no. 8, pp. 951–970, 2013.

[8] D. Droeschel, J. Stückler, D. Holz, and S. Behnke, "Towards joint attention for a domestic service robot—person awareness and gesture recognition using time-of-flight cameras," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '11)*, pp. 1205–1210, May 2011.

[9] 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.

[10] M.-Y. Wu, T.-Y. Chen, K.-Y. Chen, and L.-C. Fu, "Daily activity recognition using the informative features from skeletal and depth data," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '16)*, pp. 1628–1633, May 2016.

[11] B. Ni, G. Wang, and P. Moulin, "RGBD-HuDaAct: a color-depth video database for human daily activity recognition," in *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV Workshops '11)*, Barcelona, Spain, November 2011.

[12] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*, Cambridge University Press, Cambridge, UK, 2000.