Fisherposes for Human Action Recognition Using Kinect Sensor Data

Benyamin Ghojogh, Hoda Mohammadzade, Mozhgan Mokari

***Abstract*—This article proposes a new method for view-invariant action recognition that utilizes the temporal position of skeletal joints obtained by Kinect sensor. In this method, the actions are represented as sequences of several pre-defined poses. After pre-processing, which includes skeleton alignment and scaling, the appropriate feature vectors are obtained for recognizing and discriminating the pose of every frame by the proposed Fisherposes method. The proposed regularized Mahalanobis distance metric is used in order to recognize both the involuntary and highly made-up actions at the same time. Hidden Markov Model (HMM) is then used to classify the action related to an input sequence of poses. For taking into account the motion in the actions which are not separable by solely their temporal poses, histograms of trajectories are also proposed. The proposed action recognition method is capable of recognizing both the voluntary and involuntary actions, as well as pose-based and trajectory-based ones with a high accuracy rate. The effectiveness of the proposed method is experimented on three publicly available datasets, TST fall detection, UTKinect, and UCFKinect datasets.**

***Index Terms*—human action recognition, skeleton data, Fisher, Linear Discriminant Analysis (LDA), Hidden Markov Model (HMM), windowing, histogram, Fisherpose.**

I. Introduction

HUMAN action recognition is one of the active and important research fields in machine vision, which envisages numerous applications. For instance, it can be used in surveillance systems to control and analyze events (see Fig. 1). Interaction with computers and understanding the concepts of images semantically can be mentioned as other applications of this research field.

According to the input of system, there are two main methods for action recognition. These methods are categorized as (I) two dimensional and (II) three dimensional methods. 2D methods try to recognize the action from the two dimensional images (frames). Due to the lack of the third dimension of data, theses methods face some challenges such as occlusion of some parts of body by other parts or things, failure in detecting movements in the third dimension, etc [1], [2], [3], [4], [5], [6].

To overcome these challenges, 3D methods have been introduced. 3D reconstruction using multi-camera images is one of these approaches but it is time-consuming and not suitable for real time applications [7]. Instead, using newly introduced sensors sensing the depth of the objects to find 3D information is in interest.

Benyamin Ghojogh's e-mail: ghojogh_benyamin@ee.sharif.edu
Hoda Mohammadzade's e-mail: hoda@sharif.edu
Mozhgan Mokari's e-mail: mokari_mozhgan@ee.sharif.edu
All authors are with Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran

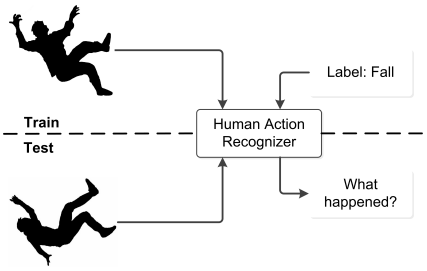


Fig. 1: A human action recognition system for fall detection.

One of these sensors is Kinect sensor that presents a depth image in addition to RGB image, using an infrared pattern. Getting this sensor commercial and availability of various software applications for it has enabled the researchers to analyze outputs of this sensor easier. By this sensor, human in the images can be tracked and his/her RGB image, depth data, and position of skeletal joints are available in multiple frames. These useful pieces of information have encouraged many researchers to investigate novel approaches in human action recognition using one, several, or a fusion of these types of information. This paper aims to use Kinect sensor in order to recognize human's actions using skeletal information. By using Kinect sensor's skeletal information as input, the proposed method models human's body and uses this information to achieve continuous action recognition.

From one perspective, action recognition can be accomplished in two ways. One way is to model actions as sequences of poses (such as the method in [30]) and the other way is to use template-based methods (such as using DTW warping [8]). We believe that the former is easier and also more successful. One of the advantages of the former category is being able to use Hidden Markov Model (HMM) for modeling actions as sequences of poses, and as HMM is robust to different speeds and lengths of performances, methods in this category are more robust. Different efforts have been already pursued for using poses with HMM, such as [30] which has modeled poses by histogram of positions of joints. However, poses are defined and modeled by a different approach in this work, i.e., poses in this work are extracted out of the high-dimensional space of the vectors of joints coordinates, and they become more separable by extracting their orthogonal discriminant directions using Fisher LDA, which is a powerful discriminant method for high-dimensional data. As is reported in experimental results section, this new definition of poses,

named as Fisherposes, outperforms the definition of poses in [30] and also [68]. In other words, poses are created more effectively resulting in better overall performance. This work also proposes/uses some other techniques in the proposed method, such as regularized Mahalanobis distance, windowing, and histogram of trajectories which are detailed in next sections. Analyzing and interpretation of Fisherposes is also one of the important points in this work. Another novelty of this work is using information of both motion and pose together in order to be able to handle various categories of actions.

This paper is organized as follows. Section II reviews related work. Section III describes the utilized datasets in this work for experiments and analysis purposes. Pre-processing is then addressed in Section IV. Section V introduces pose modeling and the selected poses in each dataset. Fisherposes is explained and analyzed in Section VI. Finding minimum distances for recognizing poses is explained in Section VII. Section VIII proposes windowing for omitting unqualified frames. Using HMM in this method is mentioned in Section IX. In Section X, histogram of trajectories are proposed in order to include motion information in the recognition pipeline. The overall structure of method is then summarized in Section XI. Effectiveness of the proposed method is verified in Section XII by experiments on the datasets. Finally Section XIII concludes the article.

II. Related Work

From a broad perspective, action recognition can be categorized into three main types, i.e., gesture recognition, simple action recognition (our focus), and activity (behavior) recognition. There are different works proposed in gesture recognition, such as [20], [21] for RGB data, [22], [23], [24] for depth data, and [25], [26] for skeleton data. Also, various works are proposed in behavior recognition, such as [9], [10], [11], [12], [13], [14] for RGB data and [15], [16], [17], [18], [19] for skeleton data. Although the proposed method might be also applicable on activity recognition, the focus of this work is on action recognition. Therefore, the related work in action recognition is reviewed here. The proposed algorithm is based on Fisher LDA, HMM, and histogram. In the following, most of the previous methods in these areas are mentioned.

A. LDA-based Methods

Fisher Linear Discriminant Analysis (LDA) is an effective method, which have been used for either dimensionality reduction, feature extraction or classification in literature [27]. There are different approaches for action recognition which used or improved LDA in their methods. In [28] and [29], both Principle Component Analysis (PCA) and LDA were used for dimension reduction and feature preparation. Mahalanobis distance was used for classification, and scale and location were canceled by making the data to have unit variance and centering the mass, respectively. In [30], middle and side hip joints were used to extract a histogram of position of other joints, which was used as feature. They reduced the dimension of feature vector using LDA. In [31], LDA was used for feature extraction. They mentioned that error due to possible inaccuracy of S_w^{-1} increases exponentially. This fact encouraged them to introduce robust LDA (RLDA) using eigenvalues of S_w. The RLDA prepared a new projection space which was used for action recognition. A novel method for feature extraction based on Canonical Correlations Analysis (CCA) in multi-linear discriminant subspace was also proposed in [32], which encoded actions as tensors. By unfolding the tensor along different tensor modes, they iteratively learned the discriminant subspace, and overcame the curse of dimensionality problem. Stepwise Linear Discriminant Analysis (SWLDA) was proposed in [33], which ran forward and backward algorithms in parallel, respectively to reduce feature space by extracting informative features and to omit irreverent features. In [34], LDA, PCA, and Locality Preserving Projections (LPP) were optimized simultaneously, resulting in Semi-supervised Discriminant analysis with Global constraint (SDG) used in action recognition. In [35], a view-invariant method was developed, and a multi-task LDA was proposed, which learns a set of linear classifiers, one for each camera view.

B. HMM-based Methods

Hidden Markov Model (HMM) is also a very effective method utilized in action recognition methods. Usually, one HMM is used for classifying each action. In [30], K-means method was used to cluster the feature vectors and create visual words. Each action was determined as a time sequence of these visual words and modeled by a HMM. In [36], orientation of body was obtained using the positions of shoulders and hip joints. Spherical angles of joints were used instead of their position, making the method robust to scale differences. In this method, an energy function was used to overcome the challenge of opposite hands or feet. Finally, HMM was used for every action. Gaussian Mixture Model (GMM) based HMM was used in [37], in which an HMM is defined for every action and the maximum likelihood output from HMMs determines the action. However, instead of using Gaussian mixture models for modeling the estimation distribution of HMM, [38] used deep neural networks because of the fact that deep nets contain many layers of features useful to predict probability distributions over states of HMM. In [39], on the other hand, angular relationship between joints were used, and HMM was used to classify the temporal motion patterns of joints. The magnitude and direction of motions were utilized in [40], [41] to generate features. They used K-means clustering for generating codebook. The minimum distance of features and codebook vectors produced symbols for HMM. A four-state HMM was considered for every action. For hands gesture trajectory recognition, [42] used Mixture of von Mises-Fisher (MvMF) Probability Density Function in HMM. In [43], action classes were represented by a discriminative multi-level Hierarchical Dirichlet Process-HMM (HDP-HMM). In their method, the model parameters of every class include transformations from a shared distribution. In other words, several shared poses were defined, which had different transition probabilities in various actions.

One of the biggest problems in using HMMs is that the number of states of HMM is hard to be optimized by trial and

error. In [44] Hierarchical Dirichlet Process Hidden Markov Model (HDP-HMM) was used, which supports an infinite number of states and automatically finds the suitable number of states. Another biggest benefit that HMM can provide is to use it for hierarchical and behavioral action recognition. In hierarchical action recognition, the top layers are related to top and longer activities, while the bottom layers are related to sub-actions from which top activities are composed. Refs. [11], [16], Layered Hidden Markov Model (LHMM) [45], variable length Markov model [46], hierarchical maximum entropy Markov model (MEMM) [47], Switching Hidden Semi-Markov Model (S-HSMM) [48], and Abstract Hidden Markov mEmory Model (AHMEM) [49] can be mentioned as examples in this category. For instance, in [11], sub-actions were determined by clustering, and in [16], a two level hierarchical HMM was used with independent Markov chains for joint positions and depth data.

C. Histogram-based Methods

Using histogram is very conventional in skeletal, depth, and RGB action recognition methods. Three-dimensional shape histogram was first introduced in [50]. In [51], the 3D histogram of the gradients (HOG3D) were used for action recognition using RGB frames. The 3D joint positions in skeletal data were also used for constructing histogram in [52]. In [30], middle and side hip joints were utilized to extract a histogram of position of other joints which were then used as the features. On the other hand, in [53], a 2D trajectory descriptor, named Histogram of Oriented Displacements (HOD), was proposed, which is fixed length, scale- and speed-invariant. In their method, the trajectory of every joint was projected on the three 2D planes xy, xz, and yz. The histogram of the three angles were formed, and the joint position offset between two consecutive frames was considered as a weight. Histogram of Oriented 4D Normals (HON4D) was also proposed in [54] for depth data. Four-dimensional histogram was applied on the 3D normal vectors of depth data along the time. In order to quantize the 4D space for constructing histogram, they considered 4D polychorons. In [55], HON4D was used alongside several other features in a fusion framework.

In [56], position offset of 3D joints were extracted as features, and bag-of-words was applied on the features. Moreover, in some methods, K-means was used with histograms. For instance, in [57], feature vectors were clustered using K-means, and bag of words was applied on the clusters. They described every sequence as a histogram of the occurrence of certain poses, resulting in Histogram of Body Poses (HBP). Another example is [58] in which several patterns were found by clustering joints orientation angles and the forward differences of these angles using K-means. For each movement, the histograms describing the frequency of occurrence of the patterns were constructed. In some methods, both temporal and spatial skeleton data were considered. In [59], the inter-skeleton motion feature and intra-skeleton structure feature were extracted out of skeleton data. These features were clustered using K-means and then histogram was applied on the quantized clustered features. In [60], relative location, velocity, and correlations of joints were found as joint features and a histogram was constructed for every joint. They used soft-binning for robustness to style variations. In soft-binning, quantized vector was added to all neighboring bins. The histograms were then normalized by the L2-norm, in order to handle different lengths of sequences.

PCA was applied on scatter matrix of the 3D depth pointclouds in [61]. By projecting each eigenvector onto 20 directions obtained from a regular 20-sided polyhedron, Histogram of Oriented Principal Components (HOPC) was obtained. In [62], the depth data was projected on the three orthogonal Cartesian planes and the differences between consecutive projection maps were thresholded in order to obtain binary maps of motion energy. The motion maps were then accumulated through the entire video sequences to generate the Depth Motion Maps (DMM). A Histogram of Oriented Gradients (HOG) was applied on each of the three DMMs. The DMM-HOG Descriptor was then constructed by concatenating the three HOGs. Histogram can also be used in behavior recognition. For example, in [12], the individual sub-actions of each person was represented as visual word histograms.

III. Description of Datasets

In this work, three different datasets are used in order to verify the effectiveness of the proposed method. These datasets are TST Fall Detection dataset [63], [64], UTKinect dataset [65], [30], and UCFKinect dataset [66], [67], which include various actions and have different applications. Merely the skeleton data of these three datasets are utilized in this work.

A. TST Dataset

The TST Fall Detection dataset [63], [64] includes two main category of actions: daily living activities and fall actions. These actions are performed by 11 different persons and performed each action three times. The daily living activities consist of sit, grasp, walk, and lying down, and the fall actions include falling front, back, side and falling backward while ends up sitting. The skeleton joints in this dataset are depicted in Fig. 2.

In this dataset, the depth and skeleton data is captured by Kinect sensor V2, which has less noise and error in extracting the exact position of joints compared to the previous version of this sensor. In addition, it provides wider field of view and higher range of operation. The main challenge of this dataset is the existence of involuntary actions such as falling in addition to their similar normal actions, which is important in elderly and patient monitoring. One of the main contributions of this work is to enhance recognition of involuntary actions whose samples do exist in this dataset.

B. UTKinect Dataset

The UTKinect dataset [65], [30] consists of 10 subjects and 10 actions, which are walk, sit down, stand up, pick up, carry, throw, push, pull, wave, and clap hands. Every action is performed twice by each person. The skeleton joints in this dataset are illustrated in Fig. 2. Two of the challenges of this

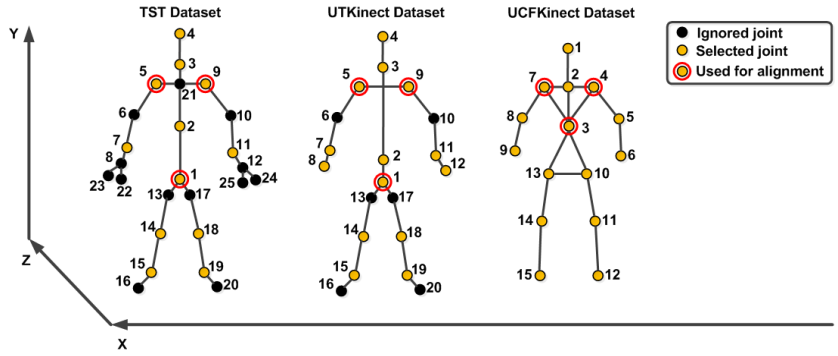


Fig. 2: Selected joints out of available joints.

dataset are different viewing angles of camera and different movement paths in the performances. These challenges are completely handled by the alignment procedure in the proposed method.

C. UCFKinect Dataset

The UCFKinect dataset [66], [67] is composed of 16 subjects, 13 males and three females, all in ages 20 to 35. There are 16 actions in this dataset, which are balance, climb ladder, climb up, duck, hop, kick, leap, punch, run, step back, step front, step left, step right, twist left, twist right, and vault. Every action is performed five times by each person, resulting in 1280 total samples. Fig. 2 shows the skeleton joints in this dataset. This dataset aims to simulate the actions in gaming applications. The challenge of this dataset is the different actions performed similarly but with different accelerations, while the acceleration data is not available. The proposed method handles this challenge by using histograms of trajectory vectors.

IV. Pre-processing

In the proposed method, the pre-processing includes skeleton alignment, scaling the skeleton, and selecting the joints, which are detailed in the following. Notice that the pre-processing functions are applied to the skeleton of every frame. Also, note that the sequence of the frames of a performance might or might not be down sampled beforehand according to the sampling rate of the dataset[1].

A. Skeleton Alignment

In skeletal-based action recognition, the position of the joints are used as inputs for the recognition system. Therefore, these positions should be somehow aligned in order to be robust against different positions and orientations of the body. Skeleton alignment includes two steps of translating hip to origin and aligning shoulders which are explained in the following.

[1]It is not recommended to down sample the frames in datasets with low number of frames, such as UTKinect in which the longest and shortest performances have respectively 114 and 5 frames.

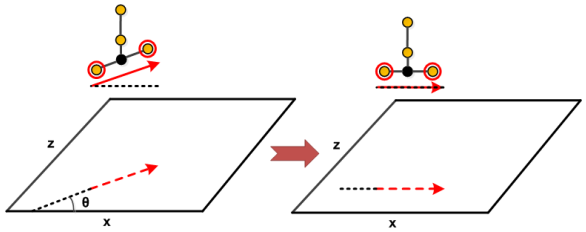


Fig. 3: Alignment of skeleton (The depicted joints are shoulders and head according to Fig. 2).

1) Hip to Origin Translation: The 3D position of a suitable joint, such as hip or spine, is used as the reference (origin) of the coordinate system in each frame, and coordinates of other joints are biased according to it (see Fig. 2). In other words, in every frame, the joints of skeleton are translated, so that the hip (or spine) falls on the origin. For the datasets without the hip joint, e.g., UCFKinect, the spine can be used interchangeably hereafter. Therefore, the positions of joints become robust to the location of body in the frame. Note that this operation omits the information of trajectory (or motion) in the action, which may cause problems in some actions. This problem and a solution to it is addressed in the later sections.

2) Aligning Shoulders: In every frame, the body may have a different orientation from the camera, which causes inaccurate recognition of skeletal pose. Hence, in every frame, the orientation of body is calculated by the left and right shoulder joints and all of the joints are rotated so that they are in a similar orientation from the camera (see Fig. 3).

To explain in more detail, the whole skeleton is rotated around the y axis, so that the projection of the vector connecting the left and right shoulders onto the xz plane (ground plane) becomes parallel to the x axis (see Fig. 3). The axes are depicted in Fig. 2.

A similar method is performed in [30] in order to align the skeleton. However, it has used left and right hip joints rather than shoulder joints. The experimental results show that using shoulder joints results in better performance.

B. Scaling Skeleton

As the height and physique of different people differ, all the skeletons should be resized and scaled to a unique size. It is important that all relative angles should be identical before and

after this step; otherwise, the poses change from their original shape. In order to reach this goal, two specific joints (such as spine and hip) are selected and their Euclidean distance d_{actual} is calculated. A target distance d_{target} (between the two joints) is also determined arbitrarily[2]. The scale factor K is then calculated as,

$$K = \frac{d_{\text{target}}}{d_{\text{actual}}}. \quad (1)$$

Afterwards, the x, y, and z coordinates of all joints are multiplied to this scale factor, resulting in the scaled skeleton to the target size, while all the angles are kept unchanged.

C. Selecting Joints

The latest version of Kinect sensor provides positions of 25 joints which are depicted in the left skeleton in Fig. 2. However, as shown in this figure, different datasets may represent different skeletons. According to the type of actions in the present datasets, the information of nearby joints are redundant, and therefore, a number of joints are selected and the rest are ignored. For TST dataset, only 12 important joints, i.e., right and left ankles, right and left knees, right and left wrists, right and left shoulders, head, spine MID, spine base (hip), and spine shoulder are considered in order to withdraw the redundant information (see the left skeleton in Fig. 2). For UTKinect dataset, however, two other joints, i.e., left and right palms, are added for more precise recognition (see the middle skeleton in Fig. 2).

The skeleton in UCFKinect dataset significantly differs, and includes 15 joints, i.e., right and left ankles, right and left knees, right and left hips, right and left wrists, right and left elbows, right and left shoulders, spine, neck (spine shoulder), and head (see the right skeleton in Fig. 2). In this dataset, position of spine joint and right and left shoulders are used for skeleton alignment.

It is worth to mention that in all three datasets, hip (spine) joint is translated to origin and therefore its position is $[0, 0, 0]^T$. Hence, this joint has redundant information in all frames, and should be removed from the skeleton after the alignment. Therefore, 11, 13, and 14 joints are selected and used respectively in TST, UTKinect, and UCFKinect datasets.

V. Pose Modeling

In this work, every action is defined as a sequence of poses of body. In other words, there exist a certain number of defined poses, and every frame represents either one of the defined poses or a middle frame (i.e., not a pose frame). Figure 4 illustrates sample actions from the three datasets, and shows their pose and middle frames. The pose frames determine the action, and the middle frames are usually estimated as one of the two surrounding poses or in some cases as another trained pose. Moreover, as shown in Balance action in this figure, some poses might be repeated in the following frames.

For every dataset, the poses should be defined in advance according to the actions of dataset, and then training instances should be sampled out from the training set of actions. The position of joints in a pose frame is considered as the input vector. The more training pose samples, the better would be the performance of the recognition system. If there are J selected joints, as every joint has three dimensions x, y, and z, it is recommended to have at least $(3 \times J)$ training samples per pose. The reason is that in order to construct an appropriate Fisher subspace (explained later), more samples than the dimension of input data are required.

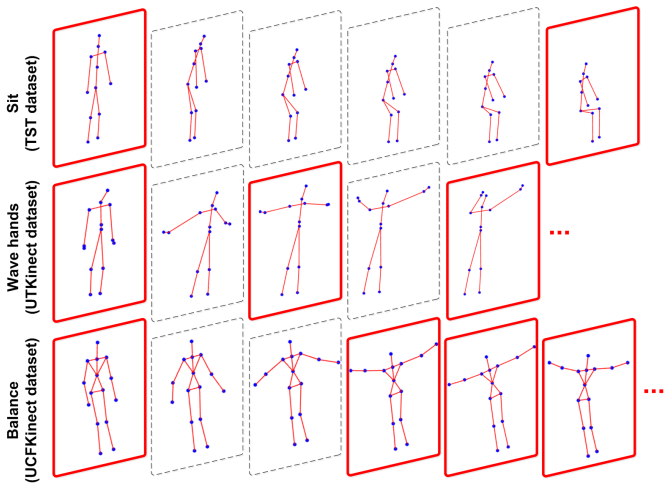


Fig. 4: An example of the composition of poses in actions. A sample action is depicted here from every dataset. The solid red frames are the poses and the dashed frames are the middle frames.

The selected poses should be orthogonal and different in information, as far as it is possible. From the statistical vantage point, the poses act as basis vectors and the actions are represented as vectors created upon these basis vectors. Also, note that the poses are selected after alignment and translation to the origin. Hence, two similar skeletons in different locations or orientations represent the same pose.

The defined poses in every dataset are listed and illustrated in Fig. 5. In our selection of poses, actions in TST dataset are made of eight poses, i.e., stand, crouching, lay back, lay front, lay side, bend, sit, and sit on ground. The actions in UTKinect dataset consist of 10 poses, which are stand, sit, bend, hands together front, start throwing, hand near shoulder, hand straight front, cross, hands up, and hands open front. UCFKinect dataset actions, however, include 12 poses, i.e., stand, cross, right hand up, left hand up, hands up, duck, kick, hand straight front, jog, curved to left, curved to right, and hands straight front.

According to the defined poses, every action is expected to be consisted of several poses. The expected compositions are reported in Table I for the three datasets. Several non-expected poses might be recognized in the actions; however, the dominant pattern of poses usually obey the patterns in this table. Also, as can be seen in this table, in UCFKinect dataset, after alignment, several actions are composed of exactly the same poses with the same order. This problem is addressed in Section X.

[2]According to the performed experiments, the target distance can slightly affect the performance, and it is set by trial and error to have the best result.

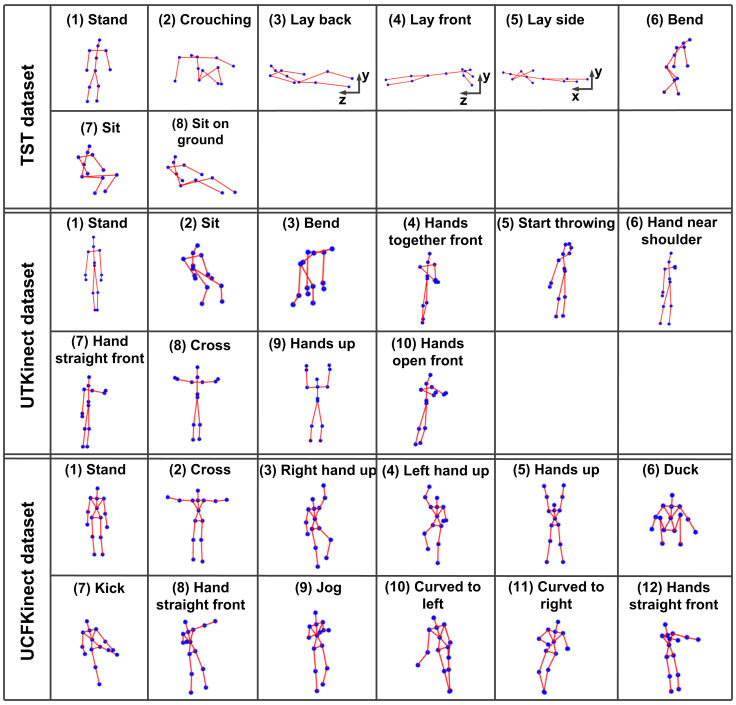


Fig. 5: An example of selected poses in TST, UTKinect, and UCFKinect datasets. Note that Kinect sensor records the skeleton as a mirror, so that the left and right side in skeleton plot is opposite to the reality.

TABLE I: Expected composition of poses in actions of datasets. Symbol $\sim$ means that a not pure pose yet close to a defined pose might occur. Moreover, symbols $\circlearrowleft$ and $\curvearrowright$ respectively denote that the action is repeated by the given pattern of poses or not. Symbol $\times$ means that the pattern (or a period of pattern) is finished.

	Action	Repetition	Expected composition of poses			
TST dataset	Sit	$\curvearrowright$	Stand	Sit	$\times$	$\times$
	Grasp	$\curvearrowright$	Stand	Bend	Stand	$\times$
	Walk	$\circlearrowleft$	Stand	Stand	$\times$	$\times$
	Lay	$\curvearrowright$	Stand	Crouching	$\sim$ Sit on ground	Lay back
	Front	$\curvearrowright$	Stand	Lay front	$\times$	$\times$
	Back	$\curvearrowright$	Stand	$\sim$ Sit on ground	Lay back	$\times$
	Side	$\curvearrowright$	Stand	Lay side	$\times$	$\times$
	End up sit	$\curvearrowright$	Stand	$\sim$ Crouching	Sit on ground	$\times$
UTKinect dataset	Walk	$\circlearrowleft$	Stand	Stand	$\times$	$\times$
	Sit down	$\curvearrowright$	Stand	Sit	$\times$	$\times$
	Stand up	$\curvearrowright$	Sit	Stand	$\times$	$\times$
	Pick up	$\curvearrowright$	Stand	Bend	Stand	$\times$
	Carry	$\circlearrowleft$	Hands together front	Hands together front	$\times$	$\times$
	Throw	$\curvearrowright$	Start throwing	Hand straight front	$\sim$ Bend	$\times$
	Push	$\curvearrowright$	Hand near shoulder	Hand straight front	$\times$	$\times$
	Pull	$\curvearrowright$	Hand straight front	Hand near shoulder	$\times$	$\times$
	Wave hands	$\circlearrowleft$	Stand	Cross	Hands up	Cross
	Clap hands	$\circlearrowleft$	Hands open front	Hands together front	Hands open front	Hands together front
UCFKinect dataset	Balance	$\circlearrowleft$	Stand	Cross	Cross	$\times$
	Climb ladder	$\circlearrowleft$	Stand	Right/left hand up	Left/hand hand up	$\times$
	Climb up	$\curvearrowright$	Stand	Hands up	Stand	$\times$
	Leap					
	Duck	$\curvearrowright$	Stand	Duck	Stand or $\times$	$\times$
	Kick	$\curvearrowright$	Stand	Kick	$\sim$ Stand	$\times$
	Punch	$\curvearrowright$	Stand	Hand straight front	$\sim$ Stand	$\times$
	Run	$\circlearrowleft$	Stand	Jog	Jog	$\times$
	Hop	$\curvearrowright$	Stand	$\sim$ Stand	Stand or $\times$	$\times$
	Step back					
	Step front					
	Step left					
	Step right					
	Twist left	$\curvearrowright$	Stand	Curved to left	$\times$	$\times$
	Twist right	$\curvearrowright$	Stand	Curved to right	$\times$	$\times$
	Vault	$\curvearrowright$	Stand	Hands straight front	$\sim$ Stand	$\times$

Note that selecting poses manually and sampling training instances for each pose out of the frames of dataset might be time-consuming. The selection and also sampling of poses can be performed automatically, although it will slightly decrease the recognition rates. A method such as [68] or [69] can be used in order to extract the training pose samples from dataset. By clustering these samples (e.g. using K-means), the appropriate poses are defined and obtained. The centers of almost all the obtained pose clusters are a subset of the complete set of manually defined poses.

VI. Fisherposes

In order to classify an input pose frame into one of the defined poses, proper features are required to be extracted. In order to extract discriminant features, Fisher Linear Discriminant Analysis (LDA) method [70], [71] is used. In Fisher LDA method, $C-1$ features are extracted from the input vectors, where C is the number of classes. Suppose that x_k is the training sample in class G_i, where x_k is constructed as,

$$x_k = [x_1, \ldots, x_J, y_1, \ldots, y_J, z_1, \ldots, z_J]^T, \tag{2}$$

and J denotes the number of selected joints. Let μ_i and μ denote the mean of i^{th} pose and total mean, respectively, and N_i be the number of training samples in class G_i. The within- (S_w) and between-class (S_b) scatter matrices are defined as,

$$S_w = \sum_{i=1}^{C} \sum_{x_k \in G_i} (x_k - \mu_i)(x_k - \mu_i)^T, \tag{3}$$

$$S_b = \sum_{i=1}^{C} N_i (\mu_i - \mu)(\mu_i - \mu)^T, \tag{4}$$

The eigenvectors of $S_w^{-1} S_b$, called here as Fisherposes, construct the Fisherpose subspace, in which the within-class is minimized while the between-class scatter is maximized [71], [72]. The feature vector for an input is obtained by projecting it onto this subspace.

Fisherposes represent the discriminative and orthogonal configuration of the joints. The first Fisherposes corresponded to the biggest eigenvalues have more information than other Fisherposes. The projections of training samples onto the first and second Fisherposes and the second and third Fisherposes are respectively depicted in figures 6 and 7 for TST dataset. Moreover, for UTKinect and UCFKinect datasets, the projections of training samples onto the first and second Fisherposes are respectively illustrated in figures 8 and 9.

For TST dataset, the corresponding joint positions for the first three Fisherposes are illustrated in Fig. 10. In this figure, solely the most motive joints, i.e., head, right and left hands, and right and left legs, are shown. As is shown in Fig. 10(a), in the first Fisherpose, hands are next to feet. Thus, by projecting the input frame into this Fisherpose, the poses with near hands and feet fall apart from the poses having far hands and feet. TST dataset includes two types of normal and abnormal actions, and this attribute does have the most important discriminative role. The same thing happens in other Fisherposes. The second Fisherpose mostly discriminates the poses with small and significant angles between forearms and

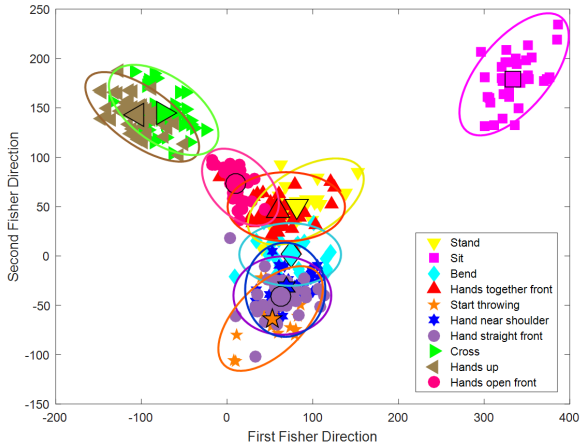


Fig. 8: Different distributions of poses in UTKinect dataset.

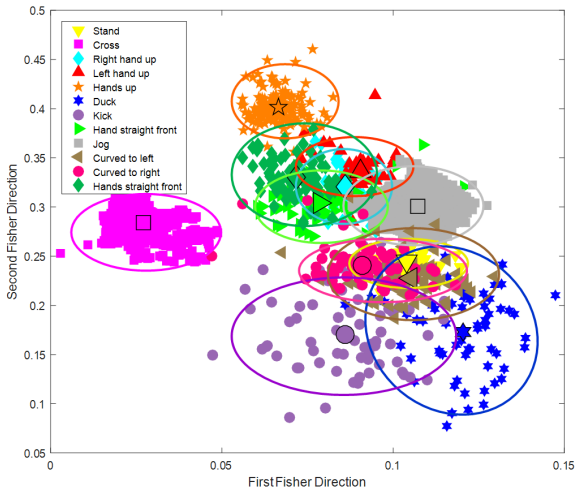


Fig. 9: Different distributions of poses in UCFKinect dataset.

legs. The third one also mostly determines whether the body is in the lay-side pose. Note that these discrimination attributes of Fisherposes are dominant but not totally limited to these mentioned features. In fact, they have other discriminative attributes, too. In figures 6 and 7, it can be seen that the different classes of poses are dominantly discriminated by the mentioned attributes in the three Fisherposes.

VII. Regularized Mahalanobis Distance

Up to this point in the method, a feature vector F is created by projecting the skeleton of the input frame onto the Fisherpose subspace. The distance of the feature vector should be calculated from the train feature vector of each defined pose. Hereafter, defined poses are called pose classes. The pose class which has minimum distance from the input feature vector, determines the pose of the input frame f, which can be written as,

$$\text{pose}(f) = \arg\min_i d(F, \widetilde{F}_i), \tag{5}$$

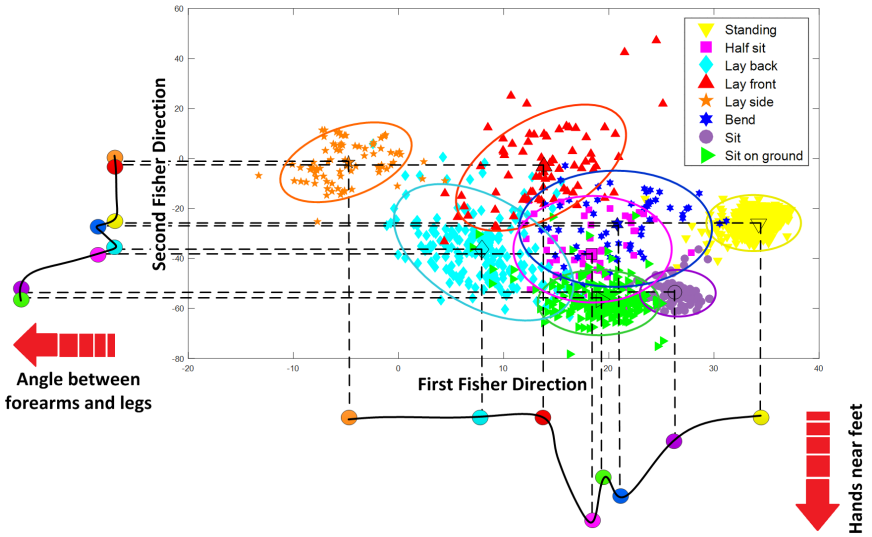


Fig. 6: Different distributions of poses in TST dataset. In bottom and left sides of this sketch, the discriminative attributes are shown. According to Fig. 10, first Fisher direction mostly determines if the hands are near feet area or not. In addition, the angular difference between forearms and legs are mostly discriminated in the second Fisher direction.

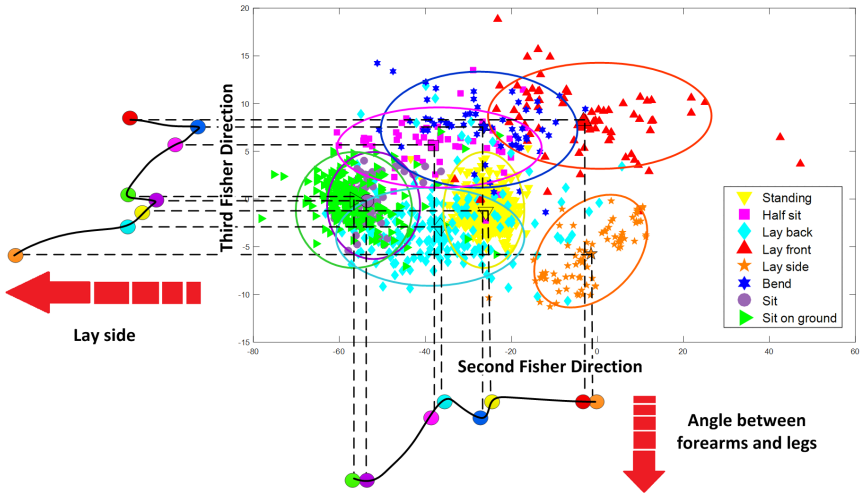


Fig. 7: Different distributions of poses in TST dataset. In bottom and left sides of this sketch, the discriminative attributes are shown. According to Fig. 10, second Fisher direction mostly determines whether the forearms are almost parallel to legs or not. Moreover, whether body is in or near lay-side pose is determined in the third Fisher direction.

where F and $\widetilde{F}_i$ respectively denote the feature vectors of the test sample and the i^{th} pose class, and d is the distance measurement function which is regularized Mahalanobis distance, proposed in the following.

As is obvious in figures 6, 7, 8, and 9, the distributions of classes are totally different, especially in TST dataset which includes involuntary actions. This different distributions are also different in various directions. Actually, the classes in which different persons perform the action more differently, have more variance. For instance, in TST dataset, these poses usually occur in involuntary actions. As an example, the pose of standing is usually performed similarly by different persons and does not have large variance. In contrast, the pose of lay-front which happens in the action of fall-front, has a large variance; because, it is an involuntary pose and therefore is performed differently even by a specific person in different performances. In addition, as is obvious in figures 6, 7, 8, and 9, the variances of distributions are different in various directions.

Mahalanobis distance possesses the advantage of considering the variance of the distribution, from which the distance of a point is calculated. However, the Mahalanobis distance itself might not be an optimum distance function. In this work, through experiments on different datasets, it was found out that depending on the type of actions in a dataset, a different distance within the range of Mahalonobis to Euclidean works best

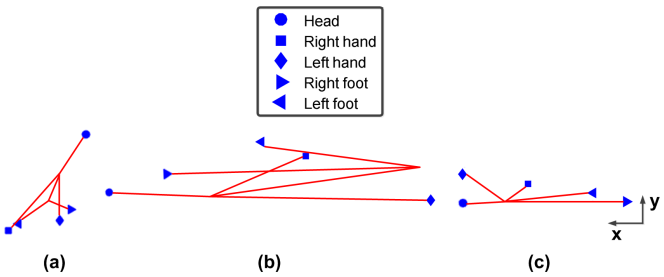


Fig. 10: Depiction of first three Fisherposes (solely most motive joints) in TST dataset. (a) First Fisher direction in which hands are near feet. (b) Second Fisher direction in which forearms and legs have significant angle difference. (c) Third Fisher direction in which body is mostly in lay-side pose.

for pose classification in each category of actions.[3] Therefore, in this work, regularized Mahalanobis distance is proposed which can be regularized by a parameter to have behavior between Mahalanobis and Euclidean distances. Suppose S and I respectively denote the covariance matrix of the feature vectors of a class and the identity matrix. The regularized Mahalanobis distance is obtained as,

$$d(F,\widetilde{F}_i) = \\ (F-\widetilde{F}_i)^T(S+\lambda I)^{-1}(F-\widetilde{F}_i)+\log(\det(S+\lambda I)), \quad (6)$$

where λ is the regularization parameter. Sweeping this parameter can result in different performances, and should be optimized by cross validation. As is obvious, the greater the λ becomes, the closer this distance gets to Euclidean distance, and the smaller the λ becomes, the closer the distance gets to Mahalonobis distance. The effect of sweeping λ is analyzed in Section XII. Notice that $\log(\det(S))$, i.e., the second term in (6) when $\lambda = 0$, is not usually included in the Mahalanobis distance in the literature; however, it is better to be considered because $\det(S)$ exists in the denominator of the Gaussian distribution formula.[4]

VIII. Windowing

Classifying a frame to one of the defined poses might be erroneous. In other words, a frame might not belong to any of the classes and be a middle frame. The distance to the nearest class itself seems to be a good measurement for the qualification of a frame. The smaller the minimum distance, the purer and more reliable would be the decision about the pose of the frame. In other words, the unqualified frames usually fall far away from the trained and correct pose after the projection onto Fisherpose subspace. Therefore, in order to remove the unqualified frames with large distances, the windowing approach is utilized.

The idea is that the unqualified frames with large enough distances should be removed. However, it might yield to very short sequences. In order to establish a balance between removing unqualified frames and having sequences with suitable lengths, windowing is used. A window with length L and sliding step T is slid over the frames of a performance. The role of the windowing is to not let the entire frames within a window be omitted.

Algorithm 1 Windowing

1: **for** $w = 1, \ldots, \lceil \frac{Q}{T} \rceil$ **do**
2: $f_{\text{window}} \leftarrow \{f_{((w-1)\times T)+1}, \ldots, f_{((w-1)\times T)+L}\}$
3: $w_{\text{empty}} \leftarrow 1$
4: $D \leftarrow 0$
5: **while** w_{empty} is 1 **do**
6: **for all** f_i in f_{window} **do**
7: Project f_i onto Fisher space
8: $d_i \leftarrow$ minimum distance
9: **if** $d_i < d_{\text{avg}} + D$ **and** $f_i > f_l$ **then**
10: Stack recognized label of f_i to sequence Se
11: $w_{\text{empty}} \leftarrow 0$
12: $f_l \leftarrow f_i$
13: **end if**
14: **end for**
15: $D \leftarrow D + S_D$
16: **end while**
17: **end for**
18: Feed sequence Se to HMM

The procedure of windowing is reported in Algorithm 1. If the sequence of performance includes Q frames, there exists $\lceil \frac{Q}{T} \rceil$ slidings (line 1 in Algorithm). Line 2 shows the frames within the window (f_{window}). Every window will remain on the frames until at least one frame of it is qualified (line 5). The frames (f_i's) within window are projected onto Fisherpose space and the minimum distance d_i is calculated (lines 7 and 8). To be qualified for acceptance, the frame should be less than a dynamic threshold (line 9). This threshold starts from the mean of means of distances of projected trained samples from the mean of their classes, denoted as d_{avg}. The threshold is then increased by step S_D while the entire frames in the window are unqualified (line 15). Finally the qualified frames (sequence Se) are fed to HMM (line 18) which is explained in the following.

IX. Hidden Markov Models

As previously mentioned, every action can be represented as a sequence of body poses. As a result, by using Hidden Markov Model (HMM), each action of dataset is modeled and recognized. A separate HMM is used for each action. HMM uses several sequences of poses as training observations and models the particular pattern that occurs during an action. For every state of HMM, there is a probability of occurence as well as probabilities of transition from that state to other states [73]. Here, the observation symbols in HMM are the classes of poses. One may refer to [74] to learn how HMM is trained and tested. The number of states of HMM for every action is found by trial and error. The method of finding the best number of states for HMMs are mentioned in Section XII-A.

[3]Note that actions can be categorized into three categories, which are involuntary, moderately constrained, and highly constrained.

[4]In this work, the second term is considered for TST dataset which includes involuntary actions, but not for UTKinect and UCFKinect datasets.

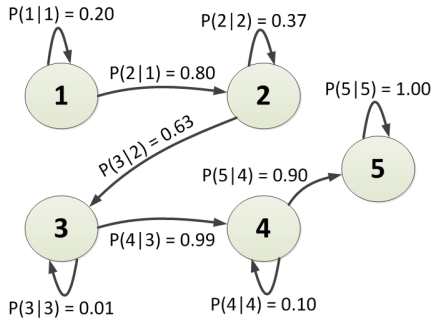


Fig. 11: A five state HMM model trained for action end up sit in TST dataset. Emission probabilities are not mentioned in the figure for the sake of brevity (e.g., for the fifth state they are $[0, 0, 0.02, 0, 0, 0, 0.16, 0.82]^T$, where the pose probabilities are sorted in the same order as in Fig. 5.

For training HMM of each action, the frame samples of training performances of that action are projected onto the trained subspace of Fisherposes and the pose of every frame is recognized. The windowing is also applied. The label of recognized pose is used as observation symbol for training HMM. For instance, if there are n poses chosen in the dataset, the observation symbols are $\{1, 2, \ldots, n\}$. The same procedure is performed in test phase. The frames in a test sample performance are projected onto the trained subspace of Fisherposes, and the recognized pose is fed to HMM as an observation symbol after applying windowing.

HMM of each action assigns a specific probability of occurrence of that action to an input sequence, and then the maximum obtained probability determines the recognized action. As an example, Fig. 11 illustrates a five state HMM along with the corresponding probabilities, trained for action end up sit in TST dataset.

It is worth noting that the use of HMM to model the sequence of poses makes the proposed method robust to speeds and lengths of an action. In other words, HMM is sensitive mainly to dynamic and pattern of poses in the action and is relatively robust to the different repeats of poses and various speeds of actions. For example, action of sequence (stand→stand→stand→stand→sit→sit) will be recognized the same as action of sequence (stand→stand→sit→sit→sit→sit→sit→sit) with a good chance. TST fall dataset and UTKinect dataset contain respectively sequences of lengths 75 frames upto 463 frames and sequences of lengths 5 frames upto 114 frames with different speeds of actions. The UCFKinect dataset also contains sequences of lengths 27 frames upto 269 frames with different speeds. In the three datasets, all of these sequences have been successfully fed into the proposed recognition system.

In addition, because of its probabilistic approach, HMM makes the method more robust to false-recognized or noisy poses. For instance, sequence (stand→ $\cdots$ →sit→ $\cdots$ →sit) determines sit action; however, a noisy sequence such as (stand→ $\cdots$ →stand→crouching→sit→ $\cdots$ →sit) has a probability close to the probability of sit action.

X. Histograms of Trajectories

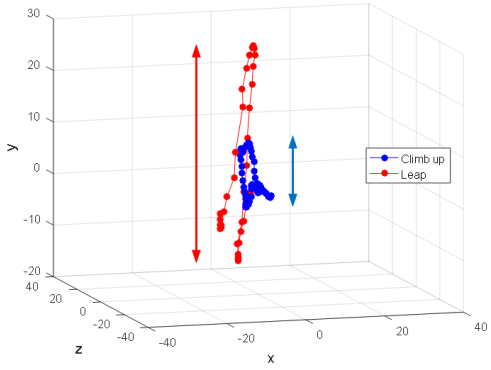
By using solely Fisherposes and HMM, the method performs well for actions which are separable by body poses. In other words, if after skeleton alignment, the actions are composed of different poses, or even the same poses but with different orders, this method works properly. Note that alignment is crucial in order to make recognition robust to position and orientation of body. However, if after alignment, the dataset includes several actions which have exactly the same poses with similar order, this method encounters problem because motion information has been omitted from the framework by alignment. In order to address this problem, histogram of trajectories is added to this method.

This part is not included in the method, unless the dataset forces according to the mentioned problem because otherwise actions are separable by poses and there is no need to this part. Hence, in this work, histograms of trajectories are utilized merely in UCFKinect dataset. In UCFKinect dataset, after alignment, all actions are separable by defined poses (see Fig. 5) except two groups of actions (see Table I). The first group includes climb up and leap. In these two actions, the skeleton jumps up, and it raises both hands while it is up in the air. Then, it comes back to ground. Thus, both these actions include poses stand, hands up, and then stand. The only difference of these two actions is that leap is performed with more acceleration and the jumping is higher.

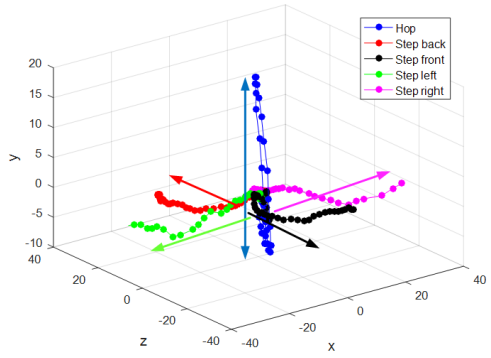
The second group consists of hop, step back, step front, step left, and step right. In hop, the skeleton jumps up with standing pose. In other four actions, the skeleton takes a step to surroundings. Therefore, all these five actions have merely pose of standing after alignment. The difference of these actions are dominantly in the direction of movement.

In order to enhance the difference of the mentioned actions in each group, the trajectories of skeleton are used. The global trajectory of skeleton can be represented by the trajectory of the hip joint as follows. Using hip joint for trajectory makes sense as this joint was removed from skeleton in previous sections because of redundancy in information (it was translated to origin), and is used back here. Note that the skeleton alignment is performed differently here than the alignment mentioned in Section IV-A. The hip in the first frame of performance is translated so that it falls on the origin. However, the hips in the following frames are translated with the translation vector of the first frame. Moreover, shoulders become parallel to x axis in the first frame while in other frames, body is rotated with rotation matrix of the first frame. By this type of alignment, motion is saved but performances are started from front orientation having no angle with z axis. The trajectory vector corresponding to each frame is the position vector of the hip in that frame. Therefore, the global trajectory is obtained as a sequence of trajectory vectors.

For every performance of the mentioned actions, a histogram is constructed as follows. The Cartesian coordinates of the trajectory vectors are first transformed to spherical coordinates $[r, \alpha, \theta]^T$, where $\alpha \in [0, 360]$ degrees and $\theta \in [0, 180]$ degrees. The histogram is 2D having bins for α and θ. Examples of these histograms can be seen in Fig. 13. The



(a) Trajectories in climb up and leap



(b) Trajectories in hop, step back, step front, step left, and step right

Fig. 12: Trajectory of motive joint (hip) in UCFKinect dataset. Notice that Kinect sensor records the skeleton as a mirror, so that the left and right side in skeleton plot is opposite to the reality.

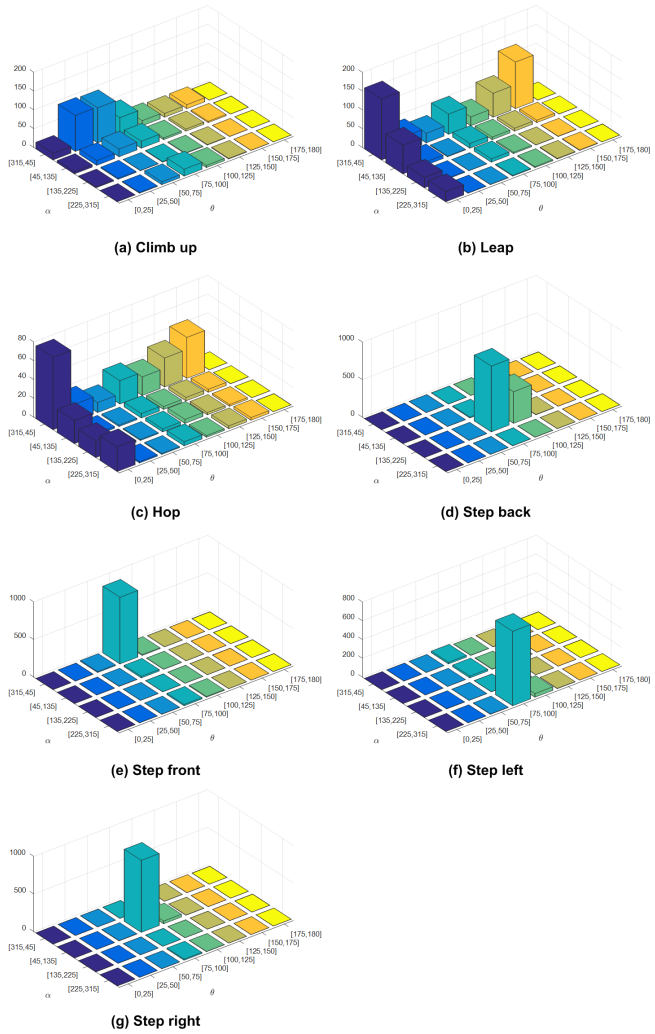


Fig. 13: Histogram of trajectories in actions climb up, leap, hop, step back, step front, step left, and step right.

magnitude of trajectory vectors, i.e. r, is used as a weight multiplied to the counts. This weight adds the information of acceleration of movement.

Figure 12 illustrates the trajectory of the hip joint in UCFKinect dataset for both of the action groups. As can be seen in this figure, in the first group, trajectory vectors of leap are much bigger than those in climb up because of the larger acceleration and higher jump. The histograms of these actions are depicted in Fig. 13. The larger bins show the more acceleration in leap. On the other hand, as can be seen in Fig. 12, the directions of motions completely differ, making the five actions distinguishable. The corresponding different histograms in Fig. 13 show that the histogram is able to distinguish these actions.

Finally, the fusion of the results of HMM and histograms is as follows. To recognize the action of an input performance, first the Fisherpose and HMM method are used to recognize the group of the action. That is, in both the training and test, all of the actions in a group are assumed as one action. After determining the group of the action, the nearest mean rule is used to classify an input sequence of trajectory vectors to one of the different actions within that group using the histogram method.

It is also worth to note that using merely histogram of trajectories for action recognition is not a good idea because it solely carries information of motion, speed, and acceleration, and does not include information of poses which is crucial for discriminating similar actions in motion and speed, such as walk and carry.

XI. Overall Structure of Proposed Method

The overall structure of the proposed method is summarized in Fig. 14. As can be seen in this figure, first, the skeleton is aligned and scaled. The joints are selected and hip is removed. In the training phase, the proper poses are defined and corresponding samples are selected for training. The Fisherpose subspace is then constructed by them. The frames of training performances are projected onto Fisherpose subspace and after windowing, the recognized frames are fed to HMM as observations. In the test phase, the same procedure is performed and HMM with the highest probability determines the action of the input sequence. According to the fact that all actions are separable by poses or not, the need to the histogram step is determined. If so, the training histograms

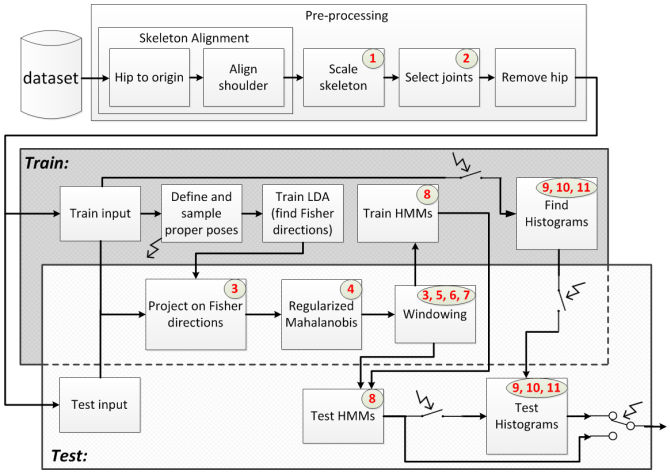


Fig. 14: The overall structure of the proposed method. The red numbers indicate the parameters set by user. These parameters are: (1) target scale in scaling body (d_{target}), (2) selected joints, (3) frame step in down sampling, (4) λ in Mahalanobis, (5) window size (L), (6) window sliding step (T), (7) step of increasing distance in windowing (S_D), (8) number of states in every HMM, (9) frame step in histograms (better to be one), (10) α bins in histograms, (11) θ bins in histograms.

TABLE II: Selected parameters for the datasets

Dataset	d_{target}	frame step	λ	L	T	S_D
TST	1500	20	1	3	1	5
UTKinect	100	5	10	1	1	–
UCFKinect	10	5	1000	1	1	–

are constructed in the train phase and used in the test phase for recognition.

In a big picture, the proposed method can be considered to be consisted of two main blocks: the Fisherpose-HMM block followed by the motion block. For an arbitrary dataset, every action should first pass through the first block and then if the input falls in a group "similar pose"actions, it should then pass through the motion block, otherwise it does not pass through this block. As a result, the motion block also exists for UTKinect and TST datasets, but no input action is recognized as being a member of a similar pose action group and therefore no input action passes through this block.

XII. Experimental Results

To examine the performance of the proposed method, TST Fall Detection dataset [63], [64], UTKinect dataset [65], [30], and UCFKinect dataset [66], [67] are used, which are explained in Section III. In the following, first, selecting the parameters are explained and analyzed followed by reporting the selected parameters for each dataset. Then, experimental results and comparison to state-of-the-art methods are reported.

A. Method Parameters

The proposed method includes a number of parameters, which are indexed in Fig. 14. These parameters are required

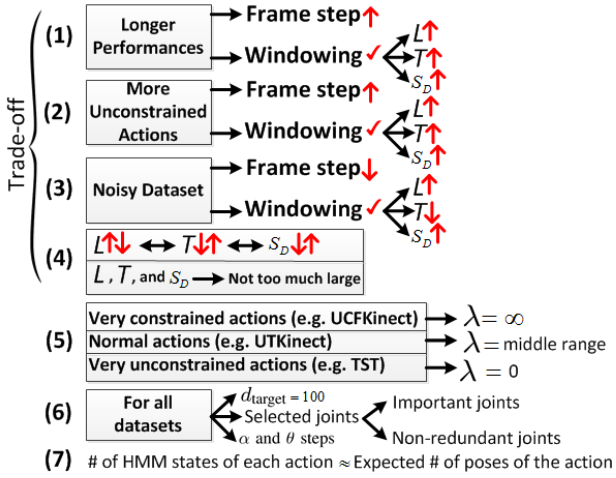


Fig. 15: Instructions on selecting parameters.

to be selected based on the type and nature of actions in the dataset and recording characteristics in order to have a better performance. One of the parameters is the selected joints which should be the important and non-redundant joints, and are shown for each of the datasets in Fig. 2. The selected parameters related to the histogram, which are α and θ steps, were previously mentioned in Section X and Fig. 13. These parameters can be fixed for all datasets because they divide spherical space properly.

Figure 15 lists the instructions for selecting parameters, where these instructions have been obtained through various observations and experiments. As shown in this figure, if dataset includes long performances or unconstrained actions (such as TST dataset), frame step should be increased and windowing should be applied with increased parameters L, T, and S_D to remove unqualified frames. For noisy dataset, however, it is better to decrease frame step to consider more frames and lessen the impact of outlier frames. In noisy dataset, windowing is needed as well but with less T because system should be more careful in checking frames and window should move more slowly. On the other hand, the larger L becomes, the smaller T and S_D are better to be chosen because larger window should be more careful in selecting qualified frames. It means that according to all characteristics of dataset, a trade-off should be applied to determine L, T, S_D, and frame step.

The selected parameters for the three datasets are reported in Table II. As seen, the selected down sampling frame step for TST dataset is much larger than that in other two datasets, which is due to the reasons mentioned in the previous paragraph. For the same mentioned reasons, windowing is useful in TST dataset, and not very promising in the other two datasets. For UTKinect and UCFKinect datasets, selected L and T are both one, which means not using the windowing step is better; therefore, S_D is not used for these datasets. Parameter d_{target} has slight impact on performance, and can be fixed for all datasets.

One of the important parameters is λ in the regularized Mahalanobis distance. Changing λ can affect the performance,

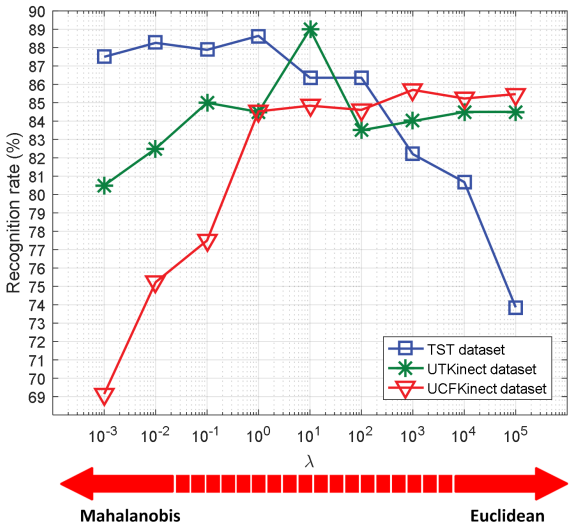


Fig. 16: Sweeping λ in regularized Mahalanobis distance for the three datasets.

which can be seen in Fig. 16. In order to find the best value for parameter λ, a validation phase is performed as follows. First, a leave-one-person-out cross validation is performed where one subject is left out as test in every fold. In every fold, again, leave-one-person-out cross validation is performed on the remaining subjects in order to find the best values of the parameter. The experiments showed that often roughly the same value is obtained as the best λ over different iterations in a dataset. The best obtained λ values for TST, UTKinect, and UCFKinect datasets are 10^0, 10^1, and 10^3 respectively. The best λ's in Figure 16, which are obtained by sweeping over different λ values on the test set, coincide with the selected parameters through validation, showing that the best selected λ's are indeed performing optimally.

As previously mentioned, according to (6), the large and small values of λ respectively result in Euclidean and Mahalanobis distances. As can be seen in Figure 16, for TST dataset, the proposed method performs better with small values of λ, i.e., Mahalanobis distance. This result makes sense because of the existence of involuntary actions, which have different and bigger distributions in Fisherposes space (see figures 6 and 7). On the other hand, for UCFKinect dataset, the method performs better with larger values of λ, i.e., Euclidean distance. This result was also expected because of the constrained gaming actions in this dataset. The best performance of the method for UTKinect dataset, however, occurs in middle range of λ because the actions in this dataset are roughly constrained, but not as much as those in UCFKinect dataset. Note that rates for $\lambda = 10^{-3} \approx 0$ and $\lambda = 10^5 \approx \infty$ in this figure are respectively demonstrating recognition rates using Mahalanobis and Euclidean distances. For selecting λ in any arbitrary dataset, actions of dataset should be checked whether they are clean and constrained or unclean and unconstrained or something in between. In this paper, utilized datasets have adequate diversity; hence, a proper dataset is experimented as a representative of each category, as explained in Fig. 16.

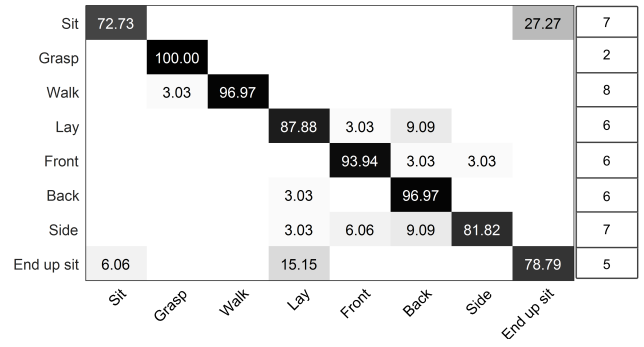
Number of states in HMM for every action can also be determined roughly by the expected number of poses of which the action is composed because it can be intuitively claimed that states of HMM are somehow representatives of poses in HMM model. Therefore, the number of states for each action depends on the type of the action itself. For selecting the best number of states for each action, initial number of states can be chosen to be the number of expected poses, and then be fine tuned using trial and error. Another approach for selecting number of HMM states, which is not based on our prior knowledge about the actions, is choosing equal numbers of states for all actions, e.g., from two to eight. The best rates for each action can then determine the number of HMM states for that action. Using either of these approaches, a look-up-table can be built up for the number of states of each action that can be used over various datasets. For the sake of brevity, the numbers of HMM states are not reported in this table, but are shown in the right columns in Fig. 17.

It is worth to mention that different values for each of the parameters rather slightly affect the performance of the proposed method, except for the λ parameter whose selection criteria was discussed in previous paragraphs. Moreover, the parameters can be fine tuned using a validation set which is different from the test set, although fine tuning for parameters other than λ does not affect the performance significantly.

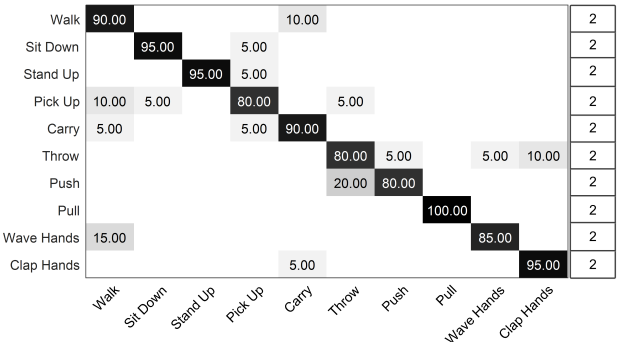
B. Experiments

1) Experiments on action versus action: Leave-one-person-out cross validation is used for experiments as discussed in Section XII-A. The confusion matrix of recognizing different actions in the three datasets are reported in Fig. 17. Accuracy rates of experimenting the proposed method on the three datasets are reported in Table III. The rates of state-of-the-art methods for the three datasets are also reported in this table.

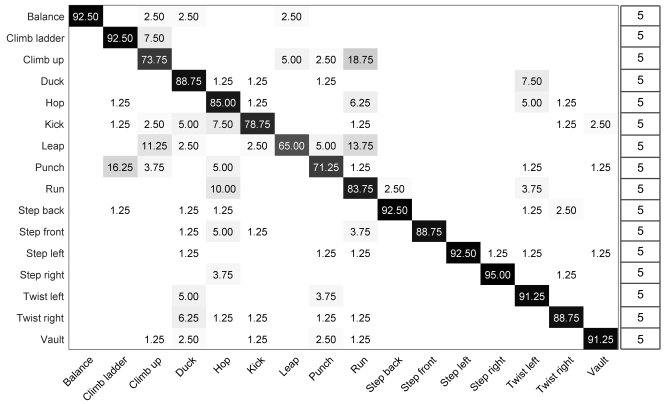
We considered TST dataset for testing other methods because we believe it is the most challenging dataset due to containing involuntary actions. For comparison we implemented methods in [30] and [68], which have introduced other definitions of poses, and we tested them on this dataset. In [30], poses are prepared using histograms of joints, probabilistic voting, and Fisher LDA, and in [68], poses are extracted using kinetic energy of skeleton and form atomic actions to model actions. On the other hand, poses in this work are extracted out of the high-dimensional space of the vectors of joints coordinates, and they become more separable by extracting Fisherposes using Fisher LDA. The mentioned various definitions of poses are demonstrated in Fig. 18. For implementing [30] and fairly comparing it with the proposed method using the TST dataset, several necessary parameter adjustments were performed. In [68], several different classifiers were used, and we used K-nearest neighbor (KNN) with $K = 1$ for this implementation because this classifier has obtained best recognition rate for new persons in their paper. According to results reported in Table III, the proposed method outperforms methods [30] and [68] on TST dataset. These outperformances show that the proposed Fisherpose definition prepares better



(a) Confusion matrix for TST dataset.



(b) Confusion matrix for UTKinect dataset.



(c) Confusion matrix for UCFKinect dataset.

Fig. 17: Confusion matrices for the utilized datasets. The columns in the right side of matrices demonstrate the number of states tuned for HMM of actions.

representation of actions in terms of poses. Moreover, it is worth to mention that methods [30] and [68] cannot be applied on datasets such as UCFKinect because of the existence of some actions which are not separable by solely poses; while the proposed method can handle this challenge as is obvious in Table III. In better words, this method benefits from combining both pose-based and motion-based approaches.

For UTKinect and UCFKinect datasets, although the proposed method does not outperform state-of-the-art methods, its performance is acceptable. The three utilized datasets have different types of features; TST dataset includes involuntary actions, UTKinect dataset consists of normal actions, and UCFKinect dataset has gaming and artificial actions. Accord-

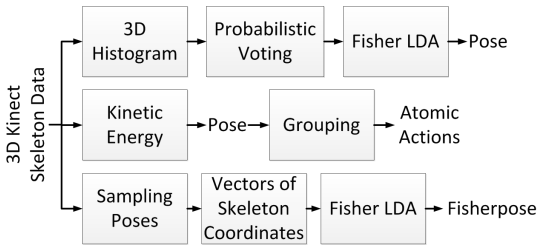


Fig. 18: Different definitions of poses in [30], [68], and this work, respectively from top to bottom.

TABLE III: Comparison of methods in TST, UTKinect, and UCFKinect datasets. Only state-of-the-art methods which have used skeleton data are reported here.

Method	TST	UTKinect	UCFKinect
[75]	–	–	98.50%
[76]	–	88.5%	97.91%
[77]	–	91.5%	99.2%
[78]	–	97.08%	–
[79]	–	98.8%	–
[30]	70.83%[1]	90.92%	×
[67]	–	–	95.94%
[80]	–	90.95%	–
[68]	84.09%[1]	–	×
Ours	88.64%	89.00%	85.70%

[1] These experiments are performed by our implementation of [30] and [68] on TST dataset.

ing to Table III, the proposed method does have acceptable recognition rates on all three datasets.

2) Experiments on fall versus normal actions: In some particular and yet important applications such as elderly surveillance, fall actions are required to be detected regardless of the type of falling. In this case, all the abnormal actions can be considered to be in one class named as the fall action and all the normal actions can be considered to be in another class. We experimented this scenario on our method, [30], and [68] for TST dataset which is a proper dataset including fall actions. Leave-one-person-out cross validation is used for this experiment as discussed in Section XII-A. In this experiment, the recognition rates for the proposed method was 90.15%, while it was obtained 77.27% and 89.39% for methods of [30] and [68], respectively. It shows the significantly better performance of the proposed method in fall-versus-normal cases in comparison to [30] and slightly better performance in comparison to [68]. In other words, it shows that Fisherposes is able to better separate diverse abnormal actions with big variances in distribution in comparison to poses in [30] and [68].

XIII. Conclusion

This article proposed a method for recognizing involuntary, normal and constrained actions. This method uses an effective framework for fusing the result of the proposed pose-based and trajectory-based approaches whenever it is required, and thus does not face any serious problem in various datasets.

Experiments showed that this method performs well on the datasets including different types of actions.

In this work, using the proposed Fisherpose method, a feature vector is created for recognizing the pose of the body in each frame. Action is recognized by constructing sequence of poses and using HMM to model sequences of each action. Therefore, the proposed method is robust to different sequence length of actions and hence to different speed of performing actions. In addition, regularized Mahalanobis distance is proposed and utilized for considering both advantages of Euclidean and Mahalanobis distances for simultaneous recognition of involuntary and voluntary actions.

XIV. Acknowledgment

This work was supported by a grant from Iran National Science Foundation (INSF). Authors thank reviewers for their precious comments which improved quality of this work.

References

[1] A. Yao, J. Gall, L. V. Gool, "Coupled action recognition and pose estimation from multiple views," *International journal of computer vision*, vol. 100, no. 1, pp. 16–37, 2012.

[2] K. Guo, P. Ishwar, J. Konrad, "Action recognition from video using feature covariance matrices," *IEEE Transactions on Image Processing*, vol. 22, no. 6, pp. 2479–2494, 2013.

[3] H. Wang, A. Klser, C. Schmid, C.L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *International journal of computer vision*, vol. 103, no. 1, pp. 60–79, 2013.

[4] J. Liu, J. Luo, M. Shah, "Recognizing realistic actions from videos in the wild," *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, IEEE*, 2009.

[5] H. Wang, A. Klser, C. Schmid, C.L. Liu, "Action recognition by dense trajectories," *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, IEEE*, 2011.

[6] J. Niebles, H. Wang, Li Fei-Fei, "Unsupervised learning of human action categories using spatial-temporal words," *International journal of computer vision*, vol. 79, no. 3, pp. 299–318, 2008.

[7] M. B. Holte, C. Tran, M.M. Trivedi, T.B. Moeslund, "Human action recognition using multiple views: a comparative perspective on recent developments," *Proceedings of the 2011 joint ACM workshop on Human gesture and behavior understanding, ACM*, 2011.

[8] Miguel Reyes, Gabriel Domnguez, and Sergio Escalera, "Featureweighting in dynamic timewarping for gesture recognition in depth data," *In Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on, IEEE*, pp. 1182–1188, 2011.

[9] Choi, Wongun, and Silvio Savarese. "A unified framework for multi-target tracking and collective activity recognition." Computer VisionECCV 2012 (2012): 215-230.

[10] Lan, Tian, et al. "Discriminative latent models for recognizing contextual group activities." IEEE Transactions on Pattern Analysis and Machine Intelligence 34.8 (2012): 1549-1562.

[11] van Kasteren, Tim LM, Gwenn Englebienne, and Ben JA Krse. "Hierarchical activity recognition using automatically clustered actions." In International Joint Conference on Ambient Intelligence, pp. 82-91. Springer Berlin Heidelberg, 2011.

[12] Kong, Yu, Yunde Jia, and Yun Fu. "Interactive phrases: Semantic descriptions for human interaction recognition." IEEE transactions on pattern analysis and machine intelligence 36, no. 9 (2014): 1775-1788.

[13] Ryoo, M. S., and J. K. Aggarwal. "Stochastic representation and recognition of high-level group activities." International journal of computer Vision 93, no. 2 (2011): 183-200.

[14] Patron-Perez, Alonso, Marcin Marszalek, Ian Reid, and Andrew Zisserman. "Structured learning of human interactions in TV shows." IEEE Transactions on Pattern Analysis and Machine Intelligence 34, no. 12 (2012): 2441-2453.

[15] Lillo, Ivan, Juan Carlos Niebles, and Alvaro Soto. "A hierarchical pose-based approach to complex action understanding using dictionaries of actionlets and motion poselets." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1981-1990. 2016.

[16] Raman, Natraj, and Stephen J. Maybank. "Activity recognition using a supervised non-parametric hierarchical HMM." Neurocomputing 199 (2016): 163-177.

[17] Lillo, Ivan, Alvaro Soto, and Juan Carlos Niebles. "Discriminative hierarchical modeling of spatio-temporally composable human activities." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 812-819. 2014.

[18] Maierdan, Maimaitimin, Keigo Watanabe, and Shoichi Maeyama. "Estimation of human behaviors based on human actions using an ANN." In Control, Automation and Systems (ICCAS), 2014 14th International Conference on, pp. 94-98. IEEE, 2014.

[19] Taha, Ahmed, Hala H. Zayed, M. Khalifa, and El-Sayed M. El-Horbaty. "Skeleton-based human activity recognition for video surveillance." International Journal of Scientific & Engineering Research 6, no. 1 (2015).

[20] Hyeon-Kyu Lee, Jin-Hyung Kim "An HMM-based threshold model approach for gesture recognition," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 21, no. 10, pp. 961–973, 1999.

[21] Andrew D. Wilson, Aaron F. Bobick "Parametric hidden markov models for gesture recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 21, no. 9, pp. 884–900, 1999.

[22] Zhou Ren, Junsong Yuan, Jingjing Meng, Zhengyou Zhang "Robust part-based hand gesture recognition using kinect sensor," *IEEE transactions on multimedia*, vol. 15, no. 5, pp. 1110–1120, 2013.

[23] Xia Liu, Kikuo Fujimura "Hand gesture recognition using depth data," *In Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on, IEEE*, pp. 529–534, 2004.

[24] Kanad K. Biswas, Saurav Kumar Basu "Gesture recognition using microsoft kinect," *In Automation, Robotics and Applications (ICARA), 2011 5th International Conference on, IEEE*, pp. 100–103, 2011.

[25] Bogdan Ionescu, Didier Coquin, Patrick Lambert, Vasile Buzuloiu "Dynamic hand gesture recognition using the skeleton of the hand," *EURASIP Journal on Advances in Signal Processing*, no. 13, pp. 2101–2109, 2005.

[26] Sait Celebi, Ali Selman Aydin, Talha Tarik Temiz, Tarik Arici "Gesture Recognition using Skeleton Data with Weighted Dynamic Time Warping," *In VISAPP*, vol. 1, pp. 620–625. 2013.

[27] Li, Tao, Shenghuo Zhu, and Mitsunori Ogihara. "Using discriminant analysis for multi-class classification: an experimental investigation." Knowledge and information systems 10, no. 4 (2006): 453-472.

[28] Weinland, Daniel, Remi Ronfard, and Edmond Boyer. "Free viewpoint action recognition using motion history volumes." Computer vision and image understanding 104, no. 2 (2006): 249-257.

[29] Carletti, Vincenzo, Pasquale Foggia, Gennaro Percannella, Alessia Saggese, and Mario Vento. "Recognition of human actions from rgb-d videos using a reject option." In International Conference on Image Analysis and Processing, pp. 436-445. Springer Berlin Heidelberg, 2013.

[30] Xia, Lu, Chia-Chih Chen, and J. K. Aggarwal. "View invariant human action recognition using histograms of 3d joints." In Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on, pp. 20-27. IEEE, 2012.

[31] Guo, Ming, and Zhelong Wang. "A feature extraction method for human action recognition using body-worn inertial sensors." In Computer Supported Cooperative Work in Design (CSCWD), 2015 IEEE 19th International Conference on, pp. 576-581. IEEE, 2015.

[32] Jia, Cheng-Cheng, Su-Jing Wang, Xu-Jun Peng, Wei Pang, Can-Yan Zhang, Chun-Guang Zhou, and Zhe-Zhou Yu. "Incremental multi-linear discriminant analysis using canonical correlations for action recognition." Neurocomputing 83 (2012): 56-63.

[33] Siddiqi, Muhammad Hameed, Rahman Ali, Md Sohel Rana, Een-Kee Hong, Eun Soo Kim, and Sungyoung Lee. "Video-based human activity recognition using multilevel wavelet decomposition and stepwise linear discriminant analysis." Sensors 14, no. 4 (2014): 6370-6392.

[34] Zhao, Xin, Xue Li, Chaoyi Pang, and Sen Wang. "Human action recognition based on semi-supervised discriminant analysis with global constraint." Neurocomputing 105 (2013): 45-50.

[35] Yan, Yan, Elisa Ricci, Ramanathan Subramanian, Gaowen Liu, and Nicu Sebe. "Multitask linear discriminant analysis for view invariant action recognition." IEEE Transactions on Image Processing 23, no. 12 (2014): 5599-5611.

[36] Papadopoulos, Georgios Th, Apostolos Axenopoulos, and Petros Daras. "Real-time skeleton-tracking-based human action recognition using kinect data." In International Conference on Multimedia Modeling, pp. 473-483. Springer International Publishing, 2014.

[37] Piyathilaka, Lasitha, and Sarath Kodagoda. "Gaussian mixture based HMM for human daily activity recognition using 3D skeleton features." In Industrial Electronics and Applications (ICIEA), 2013 8th IEEE Conference on, pp. 567-572. IEEE, 2013.

[38] Wu, Di, and Ling Shao. "Leveraging hierarchical parametric networks for skeletal joints based action segmentation and recognition." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 724-731. 2014.

[39] Huynh, Loc, Thanh Ho, Quang Tran, Thang Ba Dinh, and Tien Dinh. "Robust classification of human actions from 3D data." In Signal Processing and Information Technology (ISSPIT), 2012 IEEE International Symposium on, pp. 000263-000268. IEEE, 2012.

[40] Jalal, Ahmad, Shaharyar Kamal, and Daijin Kim. "A depth video sensor-based life-logging human activity recognition system for elderly care in smart indoor environments." Sensors 14, no. 7 (2014): 11735-11759.

[41] Jalal, Ahmad, and Shaharyar Kamal. "Real-time life logging via a depth silhouette-based human activity recognition system for smart home services." In Advanced Video and Signal Based Surveillance (AVSS), 2014 11th IEEE International Conference on, pp. 74-80. IEEE, 2014.

[42] Beh, Jounghoon, David K. Han, Ramani Durasiwami, and Hanseok Ko. "Hidden Markov Model on a unit hypersphere space for gesture trajectory recognition." Pattern Recognition Letters 36 (2014): 144-153.

[43] Raman, Natraj, and Stephen J. Maybank. "Action classification using a discriminative multilevel HDP-HMM." Neurocomputing 154 (2015): 149-161.

[44] Hu, Derek Hao, Xian-Xing Zhang, Jie Yin, Vincent Wenchen Zheng, and Qiang Yang. "Abnormal Activity Recognition Based on HDP-HMM Models." In IJCAI, pp. 1715-1720. 2009.

[45] Oliver, Nuria, Eric Horvitz, and Ashutosh Garg. "Layered representations for human activity recognition." In Proceedings of the 4th IEEE International Conference on Multimodal Interfaces, p. 3. IEEE Computer Society, 2002.

[46] Aphrodite Galata , Neil Johnson , David Hogg, Learning variable length MM of behaviour, CVIU, 2001.

[47] Sung, Jaeyong, Colin Ponce, Bart Selman, and Ashutosh Saxena. "Unstructured human activity detection from rgbd images." In Robotics and Automation (ICRA), 2012 IEEE International Conference on, pp. 842-849. IEEE, 2012.

[48] Duong, Thi V., Hung Hai Bui, Dinh Q. Phung, and Svetha Venkatesh. "Activity recognition and abnormality detection with the switching hidden semi-markov model." In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol. 1, pp. 838-845. IEEE, 2005.

[49] Nguyen, Nam Thanh, Hung Hai Bui, S. Venkatsh, and Geoff West. "Recognizing and monitoring high-level behaviors in complex spatial environments." In Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on, vol. 2, pp. II-620. IEEE, 2003.

[50] Ankerst, Mihael, Gabi Kastenmller, Hans-Peter Kriegel, and Thomas Seidl. "3D shape histograms for similarity search and classification in spatial databases." In International Symposium on Spatial Databases, pp. 207-226. Springer Berlin Heidelberg, 1999.

[51] Klaser, Alexander, Marcin Marszaek, and Cordelia Schmid. "A spatio-temporal descriptor based on 3d-gradients." In BMVC 2008-19th British Machine Vision Conference, pp. 275-1. British Machine Vision Association, 2008.

[52] Thanh, Tran Thang, Fan Chen, Kazunori Kotani, and Hoai-Bac Le. "Extraction of discriminative patterns from skeleton sequences for human action recognition." In Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), 2012 IEEE RIVF International Conference on, pp. 1-6. IEEE, 2012.

[53] Gowayyed, Mohammad Abdelaziz, Marwan Torki, Mohammed Elsayed Hussein, and Motaz El-Saban. "Histogram of Oriented Displacements (HOD): Describing Trajectories of Human Joints for Action Recognition." In IJCAI. 2013.

[54] Oreifej, Omar, and Zicheng Liu. "Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 716-723. 2013.

[55] Zhu, Yu, Wenbin Chen, and Guodong Guo. "Fusing multiple features for depth-based action recognition." ACM Transactions on Intelligent Systems and Technology (TIST) 6, no. 2 (2015): pp. 18.

[56] Lu, Guoliang, Yiqi Zhou, Xueyong Li, and Mineichi Kudo. "Efficient action recognition via local position offset of 3D skeletal body joints." Multimedia Tools and Applications 75, no. 6 (2016): 3479-3494.

[57] Shao, Ling, and Xiuli Chen. "Histogram of Body Poses and Spectral Regression Discriminant Analysis for Human Action Categorization." In BMVC, pp. 1-11. 2010.

[58] Kapsouras, Ioannis, and Nikos Nikolaidis. "Action recognition on motion capture data using a dynemes and forward differences representation." Journal of Visual Communication and Image Representation 25, no. 6 (2014): 1432-1445.

[59] Zhang, Chenyang, and Yingli Tian. "RGB-D camera-based daily living activity recognition." Journal of Computer Vision and Image Processing 2, no. 4 (2012): 12.

[60] Eweiwi, Abdalrahman, Muhammed S. Cheema, Christian Bauckhage, and Juergen Gall. "Efficient pose-based action recognition." In Asian Conference on Computer Vision, pp. 428-443. Springer International Publishing, 2014.

[61] Rahmani, Hossein, Arif Mahmood, Du Q. Huynh, and Ajmal Mian. "HOPC: Histogram of oriented principal components of 3D pointclouds for action recognition." In European Conference on Computer Vision, pp. 742-757. Springer International Publishing, 2014.

[62] Yang, Xiaodong, Chenyang Zhang, and YingLi Tian. "Recognizing actions using depth motion maps-based histograms of oriented gradients." In Proceedings of the 20th ACM international conference on Multimedia, pp. 1057-1060. ACM, 2012.

[63] TST Fall Detection dataset, http://www.tlc.dii.univpm.it/blog/databases4kinect

[64] S. Gasparrini, E. Cippitelli, E. Gambi, S. Spinsante, J. Whsln, I. Orhan, T. Lindh, "Proposal and Experimental Evaluation of Fall Detection Solution Based on Wearable and Depth Data Fusion," *ICT Innovations 2015, Advances in Intelligent Systems and Computing, Springer*, vol. 399, pp. 99–108, 2016.

[65] UTKinect-Action 3D dataset, http://cvrc.ece.utexas.edu/KinectDatasets/HOJ3D.html

[66] UCFKinect dataset, http://www.cs.ucf.edu/~smasood/datasets/UCFKinect.zip

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

[68] Shan, Junjie, and Srinivas Akella. "3D human action segmentation and recognition using pose kinetic energy." In Advanced Robotics and its Social Impacts (ARSO), 2014 IEEE Workshop on, pp. 69-75. IEEE, 2014.

[69] Ding, Wenwen, Kai Liu, Fei Cheng, and Jin Zhang. "STFC: spatio-temporal feature chain for skeleton-based human action recognition." Journal of Visual Communication and Image Representation 26 (2015): 329-337.

[70] R. A. Fisher, "The Use of Multiple Measures in Taxonomic Problems," *Ann. Eugenics*, vol. 7, pp. 179–188, 1936.

[71] T. Hastie, R. Tibshirani, J. Friedman, "The Elements of Statistical Learning, Data Mining, Inference, and Prediction," *Springer*, second edition, 2008.

[72] C. M. Bishop, "Pattern Recognition and Machine Learning," *Springer*, 2006.

[73] Ghahramani, Zoubin. "An introduction to hidden Markov models and Bayesian networks." International journal of pattern recognition and artificial intelligence 15, no. 01 (2001): 9-42.

[74] Rabiner, Lawrence R. "A tutorial on hidden Markov models and selected applications in speech recognition." Proceedings of the IEEE 77, no. 2 (1989): 257-286.

[75] Zanfir, Mihai, Marius Leordeanu, and Cristian Sminchisescu. "The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection." In Proceedings of the IEEE International Conference on Computer Vision, pp. 2752-2759. 2013.

[76] Slama, Rim, Hazem Wannous, Mohamed Daoudi, and Anuj Srivastava. "Accurate 3D action recognition using learning on the Grassmann manifold." Pattern Recognition 48, no. 2 (2015): 556-567.

[77] Devanne, Maxime, Hazem Wannous, Stefano Berretti, Pietro Pala, Mohamed Daoudi, and Alberto Del Bimbo. "3-d human action recognition by shape analysis of motion trajectories on riemannian manifold." IEEE transactions on cybernetics 45, no. 7 (2015): 1340-1352.

[78] Vemulapalli, Raviteja, Felipe Arrate, and Rama Chellappa. "Human action recognition by representing 3d skeletons as points in a lie group." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 588-595. 2014.

[79] Yang, Yanhua, Cheng Deng, Dapeng Tao, Shaoting Zhang, Wei Liu, and Xinbo Gao. "Latent Max-Margin Multitask Learning With Skelets for 3-D Action Recognition." IEEE transactions on cybernetics (2016).

[80] Theodorakopoulos, Ilias, Dimitris Kastaniotis, George Economou, and Spiros Fotopoulos. "Pose-based human action recognition via sparse representation in dissimilarity space." Journal of Visual Communication and Image Representation 25, no. 1 (2014): 12-23.