# Simultaneous Joint and Object Trajectory Templates for Human Activity Recognition from 3-D Data

Saeed Ghodsi[a], Hoda Mohammadzade[a,*], Erfan Korki[a]

[a]*Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran.*

---

**Abstract**

The availability of low-cost range sensors and the development of relatively robust algorithms for the extraction of skeleton joint locations have inspired many researchers to develop human activity recognition methods using the 3-D data. In this paper, an effective method for the recognition of human activities from the normalized joint trajectories is proposed. We represent the actions as multidimensional signals and introduce a novel method for generating action templates by averaging the samples in a "dynamic time" sense. Then in order to deal with the variations in the speed and style of performing actions, we warp the samples to the action templates by an efficient algorithm and employ wavelet filters to extract meaningful spatiotemporal features. The proposed method is also capable of modeling the human-object interactions, by performing the template generation and temporal warping procedure via the joint and object trajectories simultaneously. The experimental evaluation on several challenging datasets demonstrates the effectiveness of our method compared to the state-of-the-arts.

*Keywords:* Human Activity Recognition, RGB-D Sensors, Trajectory-based Representation, Action Template, Dynamic Time Warping (DTW), Human Object Interaction.

---

## 1. Introduction

Human activity recognition (HAR) is one of the most important research areas in computer vision. In HAR, the purpose is to utilize human movement data (e.g. an RGB video), in order to identify performed activities. Based on the complexity, human activities are usually classified into four categories: gestures, actions, interactions, and group activities [1]. Recognition of the human activities enables a broad range of applications from automated surveillance systems, patient and elderly monitoring systems, and personal assistive robotics to

*Corresponding Author

*Email addresses:* saeed.ghodsi@ee.sharif.edu (Saeed Ghodsi), hoda@sharif.edu (Hoda Mohammadzade), erfan.korki@alum.sharif.edu (Erfan Korki)

a variety of systems that involve human-computer interaction [2]. In this paper, we concentrate on the recognition of human actions as the combination of elementary body part movements.

Here we divide activity recognition challenges, into two major types. Low-level challenges are related to our data gathering method and environmental conditions. For example, view angle, size, and illumination variations, as well as occlusion, cluttering, and shadows are in this group. On the other side, high-level challenges are caused by the nature of the actions. It should be considered that individuals can perform the same action with different styles and different speeds. Even one person, depending on the situation, can perform a specific action in different ways.

Development of activity recognition methods began in the early '80s. Till recent years, research in this area was mainly focused on the recognition via 2-D video cameras. The recent availability of depth sensors with admissible precision and reasonable cost and size, motivated the computer vision community to conduct more research on the 3-D based action recognition. Aggarwal et al. [1] divided the 3-D data acquisition methods into three categories: marker-based motion capture systems, multi-view stereo images, and range sensors. The utilization of range sensors significantly alleviates the low-level challenges explained previously. Based on the extracted features from the 3-D data, Aggarwal et al. [3] classified recognition methods into five groups: features from 3-D silhouettes, features from skeletal joint locations, local spatiotemporal features, local occupancy patterns, and 3-D scene flow features.

In this paper, we propose an activity recognition system, using the 3-D location of joints and objects, extracted from the depth image sequences. We represent the human action as a set of trajectories, corresponding to the skeleton joints locations along time (Fig. 1). To make our method robust against the different styles of performing actions, we transform the joints to a human-centric coordinate system, in which, the trajectories are extracted. In this representation, human object interactions can also be modeled similarly by relative object trajectories. Then we propose a novel algorithm for the construction of template joint and object trajectories to effectively represent the actions. We also present a template-based sequence warping approach to deal with the effect of varying style, speed, and acceleration of the subjects. To consider the locality in both time and frequency domains, wavelet features are extracted from the trajectory signals. The classification results demonstrate that our proposed method is efficient and gives comparable results to the state-of-the-art approaches on several datasets.

The remainder of this paper is organized as follows. An overview of the most related methods is presented in section 2. In section 3, we first describe the preprocessing of the skeleton data, and motion representation steps. Then the template generation and temporal warping algorithms are introduced, and finally, the feature extraction and classification strategies are illustrated. Section 4 is the discussion and comparison of the experimental results of our algorithm on multiple datasets, and section 5 is the conclusion of the paper.

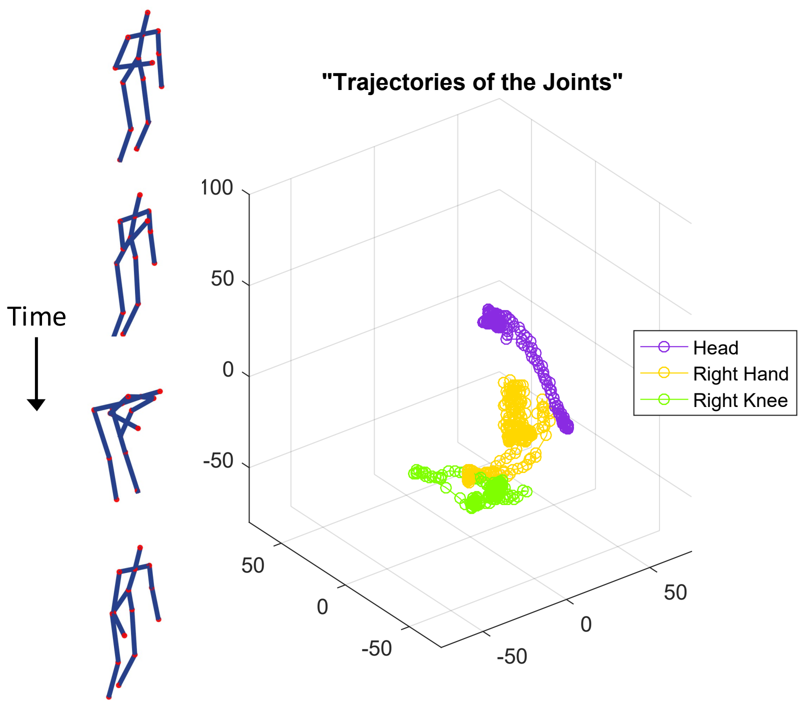



Figure 1: Joint trajectories of the ”Rinsing Mouth” action from the “CAD-60”dataset.

## 2. Related Work

In this section, a concise review of skeleton-based activity recognition methods is presented. More details are provided in [4], [5], and [6]. We also refer the interested readers to [1] and [7] for a review on RGB video-based approaches and [3], [6], and [8] for depth map-based approaches. In the following, we will review different works, from the perspective of skeletal joints representation, and the temporal modeling methodology.

In the literature, different representations are proposed for human activities. Many methods directly use the raw joint positions. Considering the location of joints as random variables, Hussein et al. [9] formed vectors to describe the actions, and then computed the covariance matrices of the vectors, to form the feature vector. Inspired by the idea of temporal pyramids, multiple covariance matrices are calculated over different windows of frames, to maintain the temporal order of the actions. Zanfir et al. [10] proposed the moving pose descriptor, which included the information of positions, as well as, speed and acceleration of the joints. In [11] the combination of feature vectors from the raw joint locations, pairwise distances between joints, and the motion of the joints are extracted and normalized. Then the Eigenjoints are generated by applying the Principle Components Analysis. To improve the recognition accuracy, Zhu et al. [12] tried to fuse skeletal joints features with spatiotemporal features. The authors used well-known image feature point detectors and descriptors, such as Histogram of Gradients (HOG), and Speeded-up Robust Features (SURF), to extract features from the depth maps. Skeletal features are extracted in the same way as [11], and after quantization with the k-means algorithm, histograms of features are fused together using the Random Forest classifier. Representation of the actions is sometimes performed by modeling the geometric relationships between the body parts. Vemulapalli et al. [13] introduced the so-called R3DG features, i.e. a family of skeleton representations. They model the human skeleton via 3-D body transformations and represent human actions as R3DG curves.

Instead of using handcrafted features, deep learning methods attempt to explain the raw data in an automatic manner. Du et al. [14] divided human skeleton into five distinct body parts and utilized a hierarchical structure of Bidirectional Recurrent Neural Networks (BRNNs) to represent the actions. In the first layer of the network, raw positions of the body parts joints were fed into the corresponding RNNs. Then the inputs of each layer were formed by a combination of the outputs of the previous layer. A fully connected layer with softmax activation was used to perform the classification. Similarly, Zhu et al. [15] proposed a three layered Long Short-Term Memory (LSTM) structure to learn human representations from the joint trajectories. Both the spatial and temporal information of the skeletal joints were utilized in [16] to train a spatiotemporal LSTM network. A Trust Gate was also proposed, to deal with the noise due to the joint location extraction. Wu and Shao [17] extracted features from the skeleton joint locations and then adopted deep belief networks to estimate the emission probabilities in Hidden Markov Models (HMMs).

Trajectory-based methods, consider an action, as a set of multiple time series representing the location of different joints over time, and extract features from the trajectories. Gupta et al. [18] introduced a motion-based descriptor to compare the Mocap data with the trajectories extracted from videos directly and generates multiple motion projections as their feature. Wei et al. [19] applied the wavelet transform and extracted features from the trajectories to address the problem of concurrent action detection. The self-similarity based descriptor, proposed by Junejo et all. [20], is an encoding mechanism for the temporal shapes of human actions observed in the videos. Experimental evaluations have shown the stability of this representation under view changes. Many methods transform the trajectories in the Euclidean space into curves in a manifold. Devanne et al. [21] proposed transforming motion trajectories into a Riemannian manifold and performing the classification using the Nearest Neighbor methods. In [22] trajectories are represented as points in the Grassmann manifold. Then the learning procedure is performed by the calculation of Control Tangents for the action clusters. Amor et al. [23] modeled trajectories on Kendalls shape manifold and introduced a new framework for the temporal alignment of the trajectories to handle the challenge of execution rate variance of the actions. Gong and Medioni [24] proposed a Spatio-Temporal Manifold (STM) to model the human joint trajectories over time. They also adapted the idea of Dynamic Time Warping to provide an algorithm for the alignment of time series under the STM model, called Dynamic Manifold Warping (DMW).

Another group of methods, try to learn dictionaries of code-words, extracted from the skeleton [25], [26]. In [27] multi-layer codebooks of key poses and atomic motions were learned using the relative orientations of body limbs. Then the action patterns were represented via the codebooks of each action, and a pattern matching algorithm was proposed to recognize the actions. Xia et al. [28] calculated Histograms of 3-D Joint locations (HOJ3D), by partitioning the space around the body of the subject to a total number of 84 bins and counting the number of joints falling in each bin. The resulting histogram represents the posture of the body. The K-means clustering algorithm is then utilized for quantization and generation of the posture vocabulary. Feeding the time domain sequences of the code-words into Hidden Markov Models (HMMs), yields statistical models representing the whole actions. Similarly, Wang et al. [29] grouped skeletal joints into five body parts and generated spatial and temporal dictionaries to represent the actions, using the K-means algorithm. Combining the group sparsity and geometry constraints, Luo et al. [30] proposed a sparse coding algorithm, to learn the dictionary, based on the relative joint locations.

Some trajectory-based approaches employ the idea of dictionary learning in the form of action templates. Muller and Roder [31] introduced the concept of motion templates to represent the actions, and then performed the recognition by a Nearest Neighbor classifier. Pairwise distances of the skeleton joints were used in [32] to learn a dictionary of motion templates. Then the Structure Streaming Skeleton (SSS) features are computed and a sparse coding approach is used for the gesture modeling. Vemulapalli et al. [33] introduced a representation for the motion trajectories, as curves in the Lie Group $SE(3) \times \cdots \times SE(3)$. To simplify the task of classification of the curves and be able to apply standard temporal modeling methods, they mapped the curves into the corresponding Lie Algebra. Then nominal curves for the actions were computed, and all the samples were warped to the curves. Following Wang et al. [34], the Fourier Temporal Pyramid (FTP) was applied, and a set of Support Vector Machines (SVMs) were adopted to perform the classification.

Due to the different discrimination power of the body joints for the recognition of actions, many methods tried to mine for the most informative joints. The proposed algorithm by Chaaraoui et al. [35] attempts to find a subset of joints, which performs the recognition task better than all joints. Dynamic Time Warping (DTW) distance of the joint location trajectories was used in [36] to measure the similarity of the action sequences. To determine the impact of each joint on the total distance function, the weighting values of joints were computed by calculating the amount of similarity of the joints trajectories in each class and dissimilarities of the trajectories between distinct classes. By determining the most informative subset of the joints for each specific action class in consecutive time segments, and then concatenating them, Ofli et al. [37] proposed a novel representation of the actions. Pairwise distances between the joints as well as Local Occupancy Patterns (LOP) around the joints were employed as features in [34]. Then Fourier Temporal Pyramid (FTP) was applied to make the representation robust against the temporal misalignment and noise. Moreover, an actionlet-based approach was introduced to mine for the most discriminative combination of the joints using the multiple kernel learning method.

In some activities, the human object interactions play an important role. In the literature, many methods have been proposed to model the human object interaction. Inspired by the idea of dividing a high-level human activity into smaller atomic actions, Wei et al. [38] introduced a hierarchical graph to represent the human pose in the 3-D space, and the motions through 1-D time. They defined an energy function, interpreted by the graph, which consists of two terms. The spatial term, includes the pose model, object model and the geometric relations between the skeleton and objects, and the temporal term includes atomic events transition and object motions. Similarly, Koppula et al. [39] aimed at jointly learning the human activities and object affordances, by defining a Markov Random Field (MRF) with two kinds of nodes, corresponding to the objects and the sub-activities. The motion and position of the objects were fed to the object node as the feature vector, and the human object interactions were modeled by the graph edges. In contrast with these works, a single layered approach was proposed in Tayyub et al. [40], to model the human object interactions, regardless of the object type. They extract qualitative and quantitative features from the objects, in the spatial and temporal domains, and apply a feature selection technique to recognize the actions efficiently. Their experiments suggested that the spatial features, i.e. the relations between the different objects in the 3-D space, have a major impact on the discrimination between distinct activities.

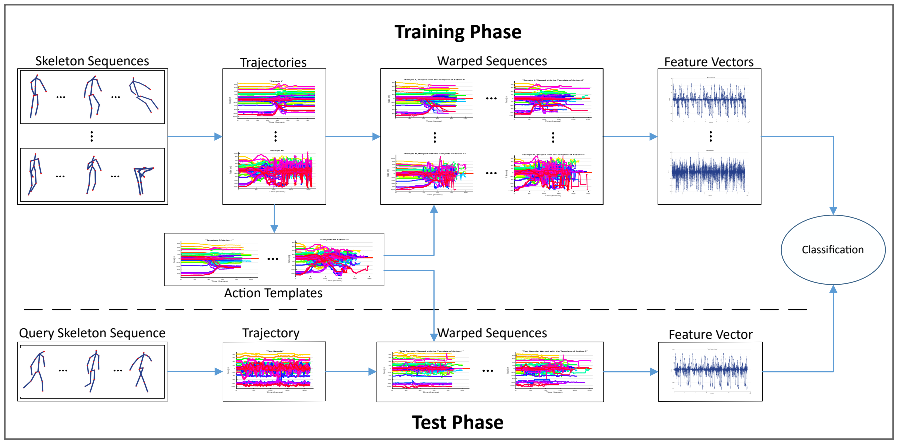



Figure 2: The general framework of the proposed approach.

## 3. Methodology

In this section, first, we explain the preprocessing of the raw 3-D data and action representation strategy. We then explain the action template generation and temporal warping steps, followed by the description of the feature generation and classification methods. An overview of our proposed framework is illustrated in Fig. 2.

### *3.1. Action Representation*

In this paper, we use a trajectory-based action representation. We model an action sample, as a set of multiple time series, each representing the variations of one coordinate of the position of one skeleton joint over time. If the actions include human-object interactions, we extract the 3-D positions of the objects and form the object trajectories. Then similar to the body joints, the object trajectories are also utilized for the action representation. Preprocessing of the raw data is usually performed to cope with the low-level challenges mentioned previously. To eliminate the effect of different positions of the subject with respect to the camera and make our method robust against the viewpoint variance, we perform a skeleton alignment procedure in each frame. For this purpose, we transform the 3-D positions of the skeleton joints, from the camera coordinates to a person-centric system by moving the hip joint of the subject to the origin, and rotating the skeleton along the $z$-axis to a predefined orientation. This geometric transformation is identical to first calculating the displacement vectors from the skeleton joints and the tracked objects to the hip joint, and then applying the same rotation to all the resulting vectors. The same translation and rotation are applied on the different skeleton joints. Some differences in the style of performing actions, such as different directions in the "walking" action, or minor body movements while "drinking water" action, will be

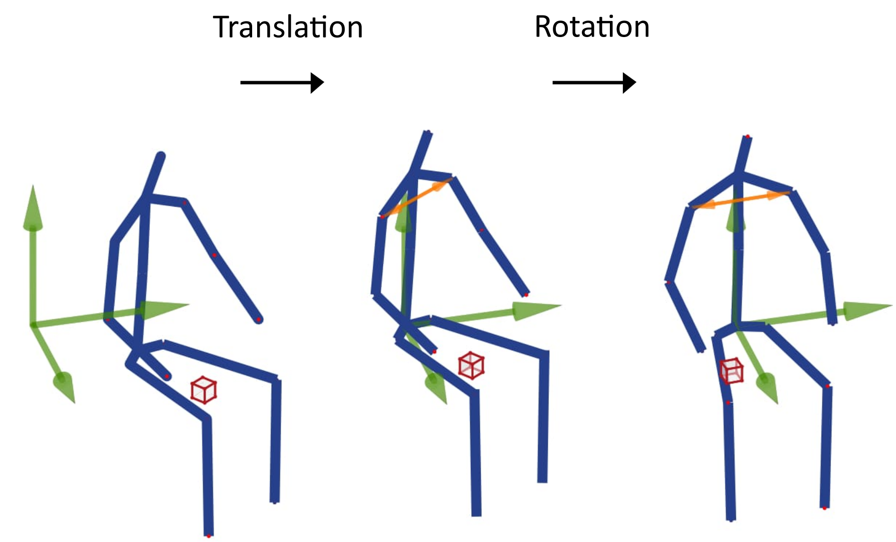



Figure 3: An illustration of the alignment procedure.

handled by performing the aforementioned geometric alignment on each frame. This alignment procedure, which is illustrated in Fig. 3, is similarly applied on all the tracked objects. More specifically, for each object, the locations of the objects 2-D bounding boxes in the RGB images are extracted by means of an off-the-shelf object detection and tracking algorithm. Then using the corresponding depth map images and the Kinect's camera calibration parameters, the real world 3-D coordinates of the object are determined along time. The extracted trajectories of the objects are used in the alignment procedure.

Let $\mathcal{J}$ and $\mathcal{O}$ be the number of tracked skeleton joints, and the maximum number of manipulated objects between the actions, respectively. Suppose $\mathfrak{S}^{(i,j)}$ be the $j$-th sample of the $i$-th action class. So the sample can be represented by the set of $\mathfrak{S}^{(i,j)} = \{\mathfrak{S}_k^{(i,j)}, k = 1, 2, \cdots, \mathcal{K}\}$, where $\mathcal{K} = (\mathcal{J} + \mathcal{O}) \times 3$ denotes the number of time series, and each $\mathfrak{S}_k^{(i,j)}$ is a single time series, corresponding to the variations of the $x$, $y$, and $z$ coordinates of one skeleton joint or tracked object in the time domain. Since the different number of objects can be present in different actions, we make the number of objects equal by placing some extra objects in the hip joint location of the subject, when needed. For example, if the actions involve at most five object manipulations, and an action has three objects, we put two extra objects in the hip joint location to make the number of time series equal. Hereafter, we consider the whole set of time series, representing an action sample, as a multidimensional signal, and name each single time series as a sub-signal. Note that the trajectories of the joints and objects are formed in the person-centric coordinates system. Then we apply a Savitzky-Golay smoothing filter [41] on the sub-signals to reduce the effect of noise, due to the depth image extraction by the Kinect sensor and the minor errors of the joints and objects position estimation. A median filter is also utilized to remove the joint position spikes.

### 3.2. *Temporal Warping*

One major issue in the action classification is the varying length and velocity of actions due to the different styles of performing actions. In the trajectory-based methods, usually Dynamic Time Warping (DTW) is utilized to deal with the temporal variations. DTW is an algorithm to find the optimal match between two given time series. Warping a sequence with another one means determining the non-linear correspondence between the time indices of the sequences, which best represents the shape similarity of them. DTW attempts to handle the deformations of the sequences in the time domain, by assigning each index in one sequence, to zero , one or more indices in the other sequence depending on the similarity between them. The output of the algorithm is the distance between the two sequences, which is defined to be the sum of the squared distances between the value of the signals at their matched indices, and also the ordered pair of the matched indices.

DTW can be employed to classify the sequences. As an example, a simple Nearest Neighbor classifier with the DTW distance measure can be adopted to determine the most similar pre-labeled action sequence to the input test sequence. Although having enough training samples, this method yields relatively good results, but the DTW algorithm is very slow in practice, even when implemented with dynamic programming techniques. Therefore comparing an input test sample with a lot of pre-labeled samples with DTW is very time-consuming and probably not appropriate for many real world applications. To cope with this challenge, we propose to warp the samples of each action, with a corresponding pre-trained action template. We first create one template for each action class in the training phase, and then in the test phase, we will use the DTW to warp the input sample merely with the templates. Thus, instead of performing DTW with many samples for each action class, we just perform the calculation with one template per action, making it much simpler.

Before explaining the template generation algorithm, we define the "mean-sample" of an action class. Let $\mathfrak{S}^{(i,j)}$, $j = 1, 2, \cdots, \mathcal{N}^i$ be the set of samples of the $i$-th action. The "mean-sample" of an action is a set of the $\mathfrak{S}_k^{(i,j)}$ sub-signals, which are most similar to the other corresponding sub-signals of this class. We find this sample by a method similar to the one proposed by Gupta and Bhavsar [42]. The method for finding the mean sample is described in Alg. 1, where $\mathcal{C}$, and $\mathcal{N}^i$ are the number of action classes, and the number of training samples for the $i$-th class respectively. In Alg. 1, the distance of the $\mathfrak{S}_k^{(i,j)}$ and $\mathfrak{S}_k^{(i,j')}$ sub-signals, is defined as the DTW distance of the two time series. The total distance value for each sub-signal of each training sample is defined as the summation of the distances from this sample to the others. The "mean-samples" are then found by minimizing the total distance values of the samples within each class. Since we calculate the sub-signals of the "mean-samples" separately, these sub-signals might come from different samples, and therefore they might

**Algorithm 1** Mean-Sample Search Algorithm

Given $\mathfrak{S}^{(i,j)}$, $\forall i, j$
**for** $i = 1, \cdots, \mathcal{C}$ **do**
  **for** $k = 1, \cdots, \mathcal{K}$ **do**
    **for** $j = 1, \cdots, \mathcal{N}^i$ **do**
      $\triangleright$ Sum up the $DTW$ distances:
      $\zeta^j \leftarrow \sum_{j'=1}^{\mathcal{N}^i} DTW(\mathfrak{S}_k^{(i,j)}, \mathfrak{S}_k^{(i,j')})$
    **end for**
    $\hat{j} \leftarrow \mathrm{argmin}_j \{\zeta^j\}$
    $\mathcal{M}_k^{(i)} \leftarrow \mathfrak{S}_k^{(i,\hat{j})}$
  **end for**
**end for**
**return** $\mathcal{M}^{(i)}$, $\forall i$

have different lengths. Experimental results demonstrate the superiority of this algorithm over other algorithms in which one of the samples are chosen as the mean sample directly.

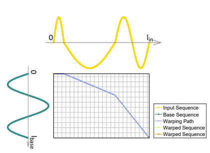
Next, we will use the "mean-samples", to achieve better representations of the action. First, we explain the algorithm for warping of a multidimensional signal with another one (Alg. 2). Let $\mathfrak{S}$ and $\mathfrak{S}'$ be two arbitrary action samples. To warp $\mathfrak{S}$ with $\mathfrak{S}'$, we perform the DTW between each pair of the corresponding sub-signals, $\mathfrak{S}_k$ and $\mathfrak{S}'_k$, $k = 1, 2, \cdots, \mathcal{K}$, and compute the optimal matching paths. Then for each $\mathfrak{S}'_k$, iterating on the indices of this time series, the value of the matched index in $\mathfrak{S}_k$ is used as the warped value of the corresponding index. If there are multiple indices assigned to one index, we'll average the values to obtain the correct warped value. It is also possible that some indices of $\mathfrak{S}_k$, wouldn't have any matching on the other side. In this case, we linearly interpolate the sequence for the missing value. All of the sub-signals are warped in this way with the corresponding sub-signals in the base multidimensional signal. At the end of this procedure, we will have the new set of sub-signals, maintaining their overall shape, while matching in the length with the base sub-signals. Some examples of sequence warping are illustrated in Fig. 4.

Now, for each action class, we create a new multidimensional signal, called "action template", as described in Alg. 3. Although templates are being generated on the basis of the corresponding "mean-samples", but, utilizing a kind of averaging method, we attempt to make them more similar to the training samples of the action. To create the template, we warp all the training samples of the class, with the "mean-sample", as explained above. Then, since all the resulting samples are the same length, we can perform a simple averaging on each index of each sub-signal, to obtain the template. An example of the template generation algorithm is presented in Fig. 5.

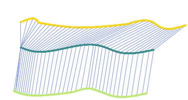
Finally, the pre-trained templates are used to warp the samples, of both training and testing sets. We warp each sample, regardless of its class, with the

**Algorithm 2** Warping Algorithm

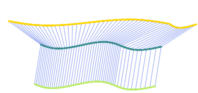
**procedure** $\textsc{Warp}(\mathfrak{S}, \mathfrak{S}')$
  **for** $k = 1, \cdots, \mathcal{K}$ **do**
    $\triangleright$ $DTW$ returns the distance and warping paths
    $[\zeta, \mathcal{P}, \mathcal{P}'] \leftarrow DTW(\mathfrak{S}_k, \mathfrak{S}'_k)$
    $i \leftarrow 1$
    $\mathcal{L} \leftarrow Len(\mathfrak{S}'_k)$
    **for** $l = 1, \cdots, \mathcal{L}$ **do**
      $\sigma \leftarrow 0$ , $n \leftarrow 0$
      **while** $\mathcal{P}'(i) = l$ **do**
        $\sigma \leftarrow \sigma + \mathfrak{S}_k[\mathcal{P}(i)]$
        $n \leftarrow n + 1$ , $i \leftarrow i + 1$
      **end while**
      **if** $n \geq 1$ **then** $\mathcal{W}_k[l] \leftarrow \dfrac{\sigma}{n}$
      **else** $\mathcal{W}_k[l] \leftarrow linear\,interpolation$
      **end if**
    **end for**
  **end for**
  **return** $\mathcal{W}$
**end procedure**



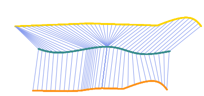
(a) Warping Path



(b) Fine Warping



(c) Ideal Warping



(d) Bad Warping

Figure 4: Examples of the sequence warping procedure.

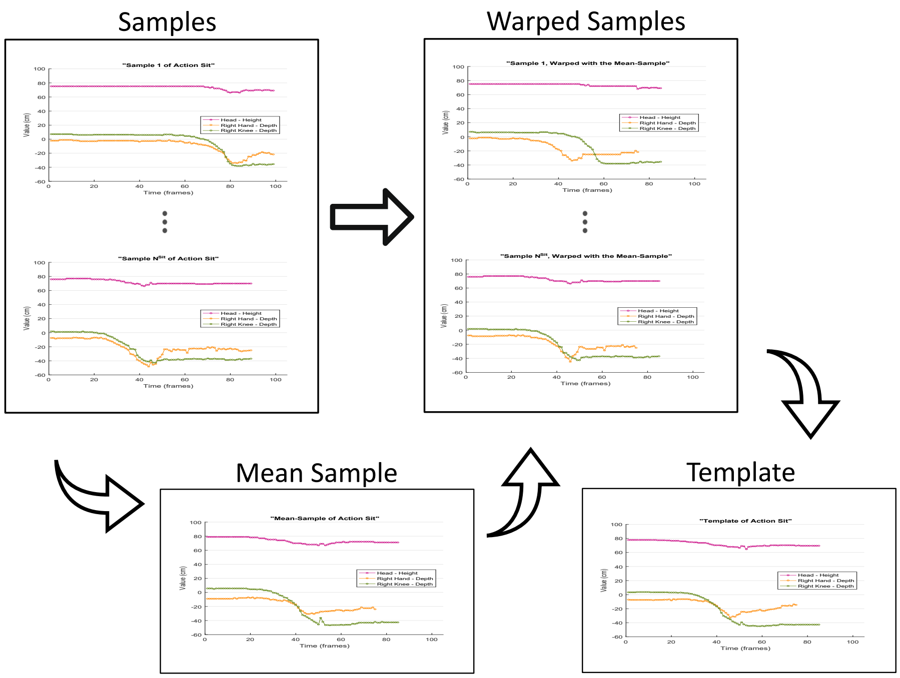



Figure 5: Illustration of the template generation algorithm for action "Sit" from the "TST Fall Detection" dataset.

---

**Algorithm 3** Template Generation Algorithm

---

**for** $i = 1, \cdots, \mathcal{C}$ **do**
  **for** $j = 1, \cdots, \mathcal{N}^i$ **do**
    $\mathfrak{S}'^{(i,j)} \leftarrow WARP(\mathfrak{S}^{(i,j)}, \mathcal{M}^{(i)})$
  **end for**
  **for** $k = 1, \cdots, \mathcal{K}$ **do**
    $\mathcal{L} \leftarrow Len(\mathcal{M}_k^{(i)})$
    **for** $l = 1, \cdots, \mathcal{L}$ **do**
      $\mathcal{T}_k^i[l] \leftarrow \dfrac{\sum_{j=1}^{\mathcal{N}^i} \mathfrak{S}_k'^{(i,j)}[l]}{\mathcal{N}^i}$
    **end for**
  **end for**
**end for**
**return** $\mathcal{T}^i$

---

templates of all actions. So if we have $\mathcal{C}$ actions in total, we will have $\mathcal{C}$ warped multidimensional signals, for each input sample.

$$\mathcal{W}^{(i,j),\nu} = WARP(\mathfrak{S}^{(i,j)}, \mathcal{T}^{\nu}), \forall i, j, \nu \quad (1)$$

This warped samples will be used together in the next step, to form the feature vectors.

### *3.3.* ***Feature Generation and Classification***

The resulting warped signals of a sample, show the matching of the sample with different templates. We performed the warping with all possible actions, to train our system the response of an input sample when warped with the positive class template and also the negative ones. To consider the localization in both time and frequency domains, we extract features from the warped multidimensional signals by the Wavelet decomposition. The Wavelet decomposition extracts features from the signal with a multilevel algorithm. At each stage, the approximation coefficients and the detail coefficients of the input signal are computed by convolving the signal with a low-pass and a high-pass filter, respectively, followed by decimation blocks. Then the approximation coefficients are fed to the next stage as input. The resulting sets of coefficients represent the low-frequency and high-frequency components of the signal, in different time scales. Here we apply the Wavelet decomposition to the sub-signals of the warped samples. Let $\mathfrak{S}$ be an arbitrary action sample. In the previous step, the warping of $\mathfrak{S}$ with different templates was performed. Suppose $\mathcal{W}^{\nu}$, $\nu = 1, \cdots, \mathcal{C}$ are the resulting warped samples. So, applying the Wavelet decomposition, we will have:

$$\mathcal{F}_k^{\nu} = Wavedec(\mathcal{W}_k^{\nu}), \forall \nu, k \quad (2)$$

The extracted coefficients from the different sub-signals are concatenated to form the feature vector. Since we have warped each specific sample with all of the templates, the extracted features from the warping results, with respect to the different templates, should also be concatenated to each other to form the total feature vector. Note that since we have warped the samples to the action templates previously, the corresponding input signals of the Wavelet decomposition filters have the same length. This causes the filter outputs, and so the total feature vectors to be meaningful for the classification purpose. An example of the temporal warping and feature vector generation algorithms is illustrated in Fig. 6.

$$\mathcal{F} = (\mathcal{F}_1^1, \cdots, \mathcal{F}_{\mathcal{K}}^1, \cdots, \mathcal{F}_1^{\mathcal{C}}, \cdots, \mathcal{F}_{\mathcal{K}}^{\mathcal{C}}) \quad (3)$$

The generated feature vectors of the training and testing samples are then used for classification purpose. Here we employ a Random Decision Forest (RDF) classifier. Random forest is an ensemble learning method that fits a number of simple and unpruned decision tree classifiers on various bootstrap samples of the data. Moreover, the split at every node of each tree is made by the best feature from among a random subset of all features. The final prediction

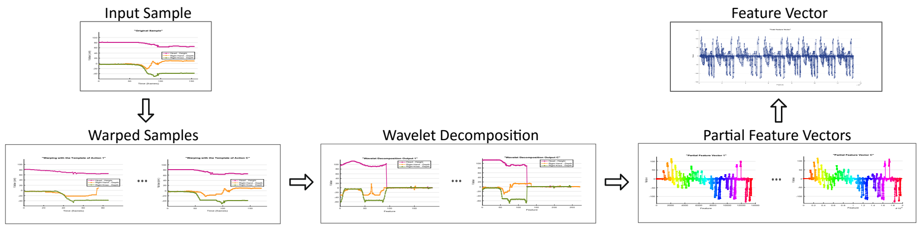



Figure 6: An example of the temporal warping and feature vector generation procedures for an arbitrary action sample.

is made by the majority vote of all trees in the forest. As each tree makes a high-variance but approximately unbiased prediction, the ensemble of trees reduces the variance and produces a relatively robust and accurate prediction.

## 4. Experiments

The Wavelet decomposition has two parameters: the Wavelet filters type, and the number of levels. In order to choose the appropriate value for this parameters, we perform a parameter tuning procedure within the training data. For this purpose, we divide the training set into two groups. Then we form the feature vectors with the different parameter values and compare the classification results between the groups. The best performing values are used for the original decomposition on the training and testing phases. We search for the best wavelet type and the number of levels between the sets of $\{Daubechies, Coiflet, Symlet\}$ and $\{1, 3, 5\}$ respectively.

In this section, we evaluate our method on five well-known datasets: Cornell Activity Datasets (CAD-60, CAD-120), UT-Kinect dataset, UCF-Kinect dataset, and TST fall detection dataset. We refer the interested readers for a review on the Kinect activity datasets to [43] and [44]. In the following, we will compare the experimental results of our method, with the state-of-the-art skeletal-based methods on each dataset. For some datasets, there may be methods using the depth and RGB modalities, achieving better results. In the cases, that k-fold cross-validation is performed, a random permutation of the subjects is considered. Then the whole process is repeated many times, and the results are averaged.

### *4.1. CAD-60 Dataset*

The CAD-60 dataset [45], is a publicly available dataset captured by the Kinect sensor. In addition to the RGB and depth map modalities, the 3-D locations of the 15 tracked skeleton joints in each frame are also available in this dataset. It consists of 12 human daily life activities, performed by four subjects in five different environments. The major issue with this dataset is the problem of handedness. Three of the subjects are right-handed, and the other one is left-handed. For example, consider the action of drinking water. Performing this

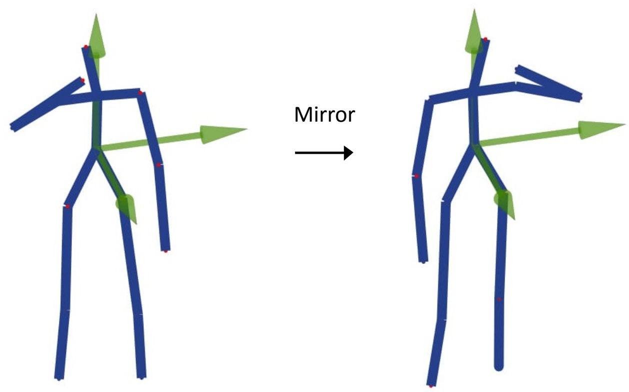



Figure 7: An illustration of the skeleton mirroring for the action "Drinking Water" from the "CAD-60"dataset.

action with the right hand, and with the left hand, will result in quite different joint trajectories, and so they will generate dissimilar feature vectors, while, they belong to the same action class. To address this issue, we adopt the well-known mirroring idea. We create a copy from each action sample in the training set, which is the mirrored version of the original sample along the bisector plane of the body. Therefore, the number of training sample will be twice, while in the test phase, merely the original samples are used. We also create two distinct templates for each action class, one for the left-handed samples and one for the right-handed ones. Then to train our system the response of the samples, to the correct and incorrect warping, we warp each action sample, regardless of its handedness, with both the templates of all classes. The final feature vectors are formed by concatenating the corresponding features of the two templates. Figures 7 and 8 give an illustration of the mirroring and warping procedures respectively.

Following [45], we use the same experimental setup. Actions are classified into five environments: office, kitchen, bedroom, bathroom, and living room. Then the Leave One Subject Out (LOSubO) cross-validation is performed for each environment, i.e. three subjects are used for the training, and the test is performed on the other one, for all possible permutations. Table 1 gives the recognition results produced by our method for the different environments. The comparison with the other methods is presented in Table 2. Except for the recent work by Zhu et al. [27], the recognition results demonstrate that our method is comparable with the state-of-the-arts.

### *4.2.* ***CAD-120 Dataset***

The CAD-120 dataset [39], is originally a high-level human activity dataset. It includes ten complex activities, performed by four subjects for three times.

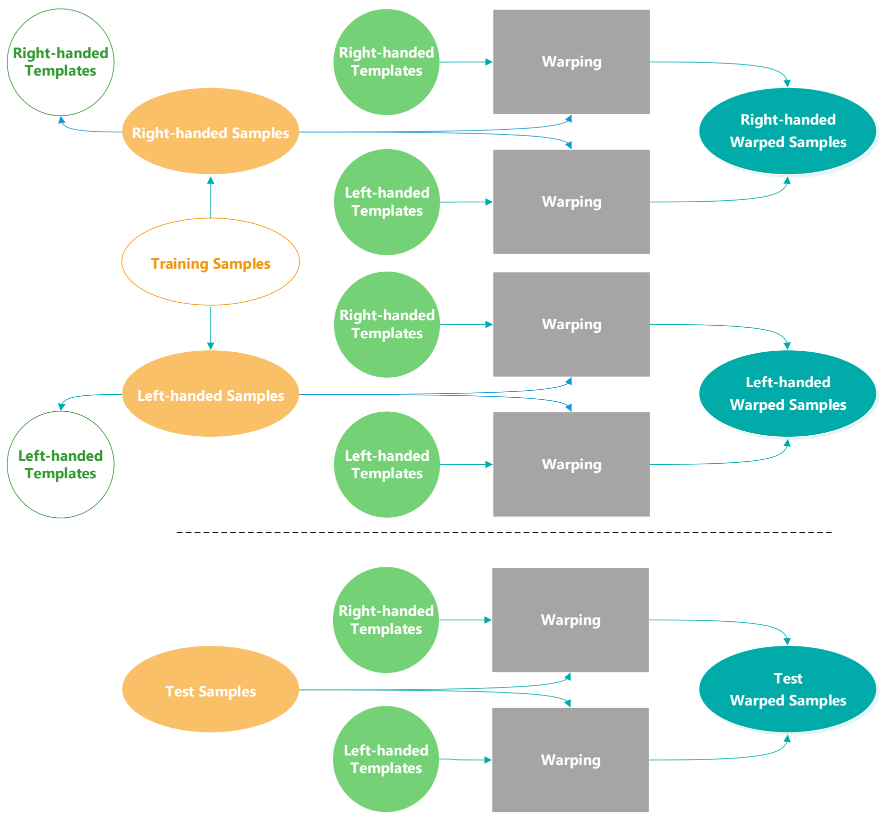



Figure 8: Warping procedure, while mirroring the samples.

Table 1: Recognition results on different environments for the “CAD-60”dataset.

| **Environment** | **Precision** | **Recall** |
|---|---|---|
| Bathroom | 100.0% | 100.0% |
| Bedroom | 91.6% | 93.3% |
| Kitchen | 93.7% | 95.0% |
| Living Room | 93.7% | 95.0% |
| Office | 87.5% | 88.7% |
| **Average** | **93.3%** | **94.4%** |

Table 2: Comparison of the different methods on the “CAD-60”dataset.

| **Method** | **Precision** | **Recall** |
|---|---|---|
| Sung et al. [45] | 67.9% | 55.5% |
| Zhu et al. [46] | 93.2% | 84.6% |
| Faria et al. [47] | 91.1% | 91.9% |
| Shan and Akella [48] | 93.8% | 94.5% |
| Gaglio et al. [49] | 77.3% | 76.7% |
| Parisi et al. [50] | 91.9% | 90.2% |
| Cippitelli et al. [51] | 93.9% | 93.5% |
| Zhu et al. [27] | **97.4**% | **95.8**% |
| **our method** | **93.3%** | **94.4%** |

Each action consists of a sequence of atomic activities called sub-activities. Our motivation to choose the CAD-120 dataset was the importance of the object manipulations in the activities of this dataset. All of the ten high-level activities include human object interactions. In some cases, e.g. the stacking objects and unstacking objects, the discrimination between the actions is significantly caused by the objects. In this dataset, an object tracking algorithm was applied on the RGB images of the frames of all the samples, and the 2D locations of the objects bounding boxes were specified. We have used the bounding boxes to extract the 3-D location of the objects using the corresponding depth map images.

Although our method does not concentrate on the high-level activities, the evaluation results on this dataset demonstrate comparable performance of our method with the state-of-the-arts. The confusion matrix is presented in Fig. 9. As this figure shows, the main trouble with this dataset is about confusing the activities “stacking objects”with “unstacking objects”, “microwaving food”with “cleaning objects”, and “arranging objects”with “picking objects”, which are very similar. Comparison of our method with the state-of-the-arts is shown in Table 3. In the dataset, the ground-truth temporal segmentation of the actions


Figure 9: Confusion matrix for the "CAD-120"dataset.

Table 3: Comparison of the high-level recognition accuracies of the different methods on the "CAD-120"dataset.

| **Method** | **Without ground-truth** | **With ground-truth** |
|---|---|---|
| Koppula et al. [39] | 80.6% | 84.7% |
| Hu et al. [52] | 87.0% | - |
| Tayyub et al. [40] | **95.2%** | - |
| Taha et al. [53] | - | 94.4% |
| Koppula and Saxena [54] | 83.1% | 93.5% |
| **our method** | **90.1%** | - |

was provided. Some hierarchical methods have used this segmentation data to improve their results. Since our method recognizes the high-level actions in one stage, we have not used this data.

Table 4: Comparison of the different methods on the "UT-Kinect"dataset, using the Cross Subject setting.

| **Method** | **Accuracy** |
|---|---|
| Vemulapalli et al. [33] | **97.0%** |
| Antunes et al. [57] | 95.1% |
| Gupta and Bhavsar [42] | 96.0% |
| **our method** | **96.8%** |

### *4.3.* ***UT-Kinect Dataset***

The UT-Kinect dataset was introduced in [28]. The dataset consists of ten actions: walk, sit down, stand up, pick up, carry, throw, push, pull, wave and clap hands. Each action is performed twice by ten different subjects in a lab environment, and 20 skeleton joints are tracked in each frame. The relatively high within-class variance is a considerable challenge with this dataset. The different actions of this dataset are performed continuously by each subject, and the temporal segmentation is manually provided.

To be comparable with the previous works, we have tested our algorithm using 2-fold cross subject validation setting, i.e. for a random permutation of the subjects, half of them were used for the training and the remaining for testing, and then vice versa. The comparison of our method with the state-of-the-arts is presented in Table 4. It should be mentioned that Xia et al. [28], and Cippitelli et al. [51] had reported 90%, and 95.1% recognition accuracies respectively, using the Leave One Sequence Out (LOSeqO) experimental setup. Also, Liu et al. [55] and Yang et al. [56] had achieved the 95.5% and 98.8% accuracies, adopting the Leave One Subject Out (LOSubO) and 10-fold cross-validation settings, respectively. Since these experimental settings are rather easier in comparison with the 2-fold method, we have reported in Table 4 only the methods which have adopted the 2-fold setting.

### *4.4.* ***UCF-Kinect Dataset***

Ellis et al. [58] presented the UCF-Kinect dataset to evaluate their latency-aware learning algorithm, which focuses on reducing the recognition latency. The dataset was captured using a Kinect sensor with the OpenNI platform, which provides the 3-D coordinates of the 15 skeleton joints. It contains 16 short actions, performed by 16 subjects for five times. Similar to the experimental setting in [58], we use the 4-fold cross subject validation as evaluation protocol for this dataset. The comparison with the other methods is shown in Table 5. Slama et al. [22] reported the 97.9% recognition accuracy, for a 0.7 and 0.3 split on the 1280 samples of the dataset, for the training and testing sets. Also, Jiang et al. [59] had achieved the 98.7% accuracy, adopting the 2-fold setting on the samples.

Table 5: Comparison of the different methods on the "UCF-Kinect"dataset.

| **Method** | **Accuracy** |
|---|---|
| Zanfir et al. [10] | 98.5% |
| Kerola et al. [60] | 98.8% |
| Yang et al. [11] | 97.1% |
| Beh et al. [61] | **98.9**% |
| Ding et al. [62] | 98.0% |
| Lu et al. [63] | 97.6% |
| **our method** | **97.9%** |

### *4.5. TST Fall Detection Dataset*

This dataset was originally collected by Gasparrini et al. [64] as a part of a study on the human fall event detection problem. They aimed at using the fusion of camera and wearable sensors to detect the fall event. The dataset was collected using the Microsoft Kinect v2 and the Inertial Measurement Unit (IMU) sensors. In this dataset two groups consisting of four daily living actions and four fall actions were performed by 11 subjects for three times. Although the wearable sensors provide very valuable data, we don't use this modality in our work and perform the recognition just utilizing the tracked skeleton joints data. Same as [64], we evaluated our method with the Leave One Subject Out cross-validation (LOSubO) setting. The average accuracy of our method for all the activities is 92.8%. Note that in [64] the 99% recognition accuracy is reported using the multiple modalities, including the wearable sensors, and so the results are not comparable. The confusion matrix of our method is illustrated in Fig. 10.

## 5. Conclusion

In this paper, we have developed a trajectory-based activity recognition system. We represented a human action as a set of time series corresponding to the normalized coordinates of the skeleton joints. Our representation is also able to simultaneously model the interaction between human and objects in the scene. Then we introduced an algorithm to effectively construct templates for joint and object trajectories. Also, a DTW-based warping procedure was proposed to alleviate the effects of variations in the styles of performing actions. The wavelet filters were utilized to extract meaningful features from the signals, and the classification was performed by the Random Decision Forests. The experimental evaluation of the proposed method on several public datasets yielded comparable performance to the state-of-the-arts. Although our proposed method works well on the recognition of simple and short actions, the template-based approaches have problems with the more complex activities. Representing the activities which consist of multiple simple sub-actions using one unique template, will not

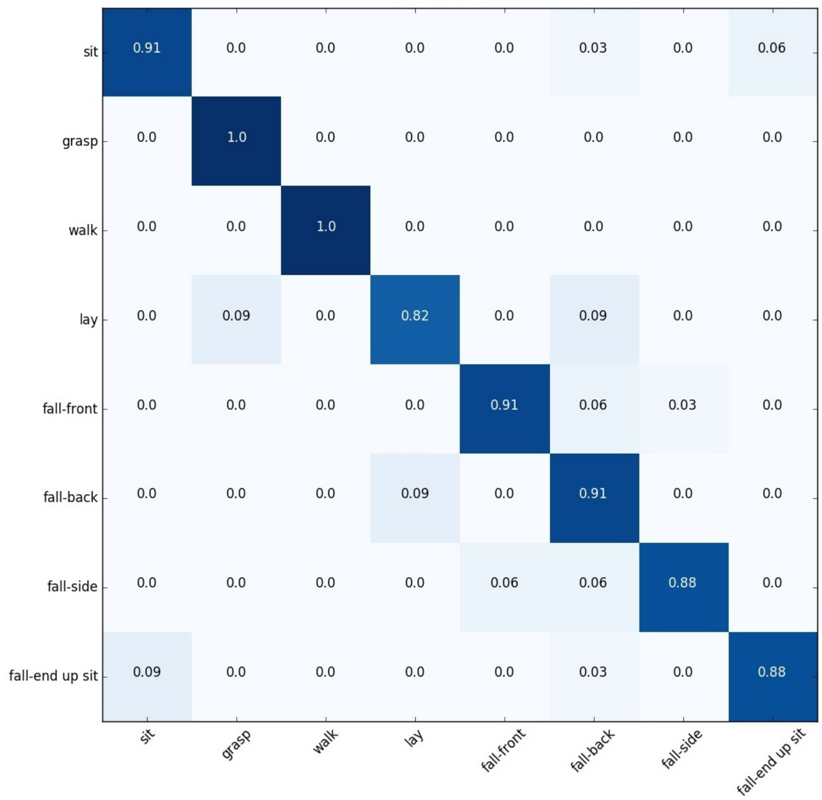



Figure 10: Confusion matrix for the “TST Fall Detection”dataset.

have good recognition results, due to their nature. So next we plan to apply modifications to our method to make it usable for the complex human activities.

## Acknowledgment

This work was supported by a grant from Iran National Science Foundation (INSF).

## References

[1] J. K. Aggarwal, M. S. Ryoo, Human activity analysis: A review, ACM Computing Surveys (CSUR) 43 (3) (2011) 16.

[2] R. Lun, W. Zhao, A survey of applications and human motion recognition with microsoft kinect, International Journal of Pattern Recognition and Artificial Intelligence 29 (05) (2015) 1555008.

[3] J. K. Aggarwal, L. Xia, Human activity recognition from 3d data: A review, Pattern Recognition Letters 48 (2014) 70–80.

[4] F. Han, B. Reily, W. Hoff, H. Zhang, Space-time representation of people based on 3d skeletal data: A review, arXiv preprint arXiv:1601.01006.

[5] L. L. Presti, M. La Cascia, 3d skeleton-based human action classification: a survey, Pattern Recognition 53 (2016) 130–147.

[6] M. Ye, Q. Zhang, L. Wang, J. Zhu, R. Yang, J. Gall, A survey on human motion analysis from depth data, in: Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications, Springer, 2013, pp. 149–187.

[7] D. Weinland, R. Ronfard, E. Boyer, A survey of vision-based methods for action representation, segmentation and recognition, Computer vision and image understanding 115 (2) (2011) 224–241.

[8] L. Chen, H. Wei, J. Ferryman, A survey of human motion analysis using depth imagery, Pattern Recognition Letters 34 (15) (2013) 1995–2006.

[9] M. E. Hussein, M. Torki, M. A. Gowayyed, M. El-Saban, Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations., in: IJCAI, Vol. 13, 2013, pp. 2466–2472.

[10] M. Zanfir, M. Leordeanu, C. Sminchisescu, The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 2752–2759.

[11] X. Yang, Y. Tian, Effective 3d action recognition using eigenjoints, Journal of Visual Communication and Image Representation 25 (1) (2014) 2–11.

[12] Y. Zhu, W. Chen, G. Guo, Fusing spatiotemporal features and joints for 3d action recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2013, pp. 486–491.

[13] R. Vemulapalli, F. Arrate, R. Chellappa, R3dg features: Relative 3d geometry-based skeletal representations for human action recognition, Computer Vision and Image Understanding 152 (2016) 155–166.

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

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

[16] J. Liu, A. Shahroudy, D. Xu, G. Wang, Spatio-temporal lstm with trust gates for 3d human action recognition, in: European Conference on Computer Vision, Springer, 2016, pp. 816–833.

[17] D. Wu, L. 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, 2014, pp. 724–731.

[18] A. Gupta, J. Martinez, J. J. Little, R. J. Woodham, 3d pose from motion for cross-view action recognition via non-linear circulant temporal encoding, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 2601–2608.

[19] P. Wei, N. Zheng, Y. Zhao, S.-C. Zhu, Concurrent action detection with structural prediction, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 3136–3143.

[20] I. N. Junejo, E. Dexter, I. Laptev, P. Perez, View-independent action recognition from temporal self-similarities, IEEE transactions on pattern analysis and machine intelligence 33 (1) (2011) 172–185.

[21] M. Devanne, H. Wannous, S. Berretti, P. Pala, M. Daoudi, A. Del Bimbo, 3-d human action recognition by shape analysis of motion trajectories on riemannian manifold, IEEE transactions on cybernetics 45 (7) (2015) 1340–1352.

[22] R. Slama, H. Wannous, M. Daoudi, A. Srivastava, Accurate 3d action recognition using learning on the grassmann manifold, Pattern Recognition 48 (2) (2015) 556–567.

[23] B. B. Amor, J. Su, A. Srivastava, Action recognition using rate-invariant analysis of skeletal shape trajectories, IEEE transactions on pattern analysis and machine intelligence 38 (1) (2016) 1–13.

[24] D. Gong, G. Medioni, Dynamic manifold warping for view invariant action recognition, in: Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE, 2011, pp. 571–578.

[25] R. Chaudhry, F. Ofli, G. Kurillo, R. Bajcsy, R. Vidal, Bio-inspired dynamic 3d discriminative skeletal features for human action recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2013, pp. 471–478.

[26] C. Wu, J. Zhang, S. Savarese, A. Saxena, Watch-n-patch: Unsupervised understanding of actions and relations, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 4362–4370.

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

[28] L. Xia, C.-C. Chen, J. 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, IEEE, 2012, pp. 20–27.

[29] C. Wang, Y. Wang, A. L. Yuille, An approach to pose-based action recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 915–922.

[30] J. Luo, W. Wang, H. Qi, Group sparsity and geometry constrained dictionary learning for action recognition from depth maps, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 1809–1816.

[31] M. Müller, T. Röder, Motion templates for automatic classification and retrieval of motion capture data, in: Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation, Eurographics Association, 2006, pp. 137–146.

[32] X. Zhao, X. Li, C. Pang, X. Zhu, Q. Z. Sheng, Online human gesture recognition from motion data streams, in: Proceedings of the 21st ACM international conference on Multimedia, ACM, 2013, pp. 23–32.

[33] R. Vemulapalli, F. Arrate, R. Chellappa, Human action recognition by representing 3d skeletons as points in a lie group, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 588–595.

[34] J. Wang, Z. Liu, Y. Wu, J. Yuan, Mining actionlet ensemble for action recognition with depth cameras, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 1290–1297.

[35] A. A. Chaaraoui, J. R. Padilla-López, P. Climent-Pérez, F. Flórez-Revuelta, Evolutionary joint selection to improve human action recognition with rgb-d devices, Expert systems with applications 41 (3) (2014) 786–794.

[36] M. Reyes, G. Domínguez, S. Escalera, Featureweighting in dynamic timewarping for gesture recognition in depth data, in: Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on, IEEE, 2011, pp. 1182–1188.

[37] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, R. Bajcsy, Sequence of the most informative joints (smij): A new representation for human skeletal action recognition, Journal of Visual Communication and Image Representation 25 (1) (2014) 24–38.

[38] P. Wei, Y. Zhao, N. Zheng, S.-C. Zhu, Modeling 4d human-object interactions for event and object recognition, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 3272–3279.

[39] H. S. Koppula, R. Gupta, A. Saxena, Learning human activities and object affordances from rgb-d videos, The International Journal of Robotics Research 32 (8) (2013) 951–970.

[40] J. Tayyub, A. Tavanai, Y. Gatsoulis, A. G. Cohn, D. C. Hogg, Qualitative and quantitative spatio-temporal relations in daily living activity recognition, in: Asian Conference on Computer Vision, Springer, 2014, pp. 115–130.

[41] A. Savitzky, M. J. Golay, Smoothing and differentiation of data by simplified least squares procedures., Analytical chemistry 36 (8) (1964) 1627–1639.

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

[43] M. Firman, Rgbd datasets: Past, present and future, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2016, pp. 19–31.

[44] J. Zhang, W. Li, P. O. Ogunbona, P. Wang, C. Tang, Rgb-d-based action recognition datasets: A survey, Pattern Recognition 60 (2016) 86–105.

[45] J. Sung, C. Ponce, B. Selman, A. Saxena, Unstructured human activity detection from rgbd images, in: Robotics and Automation (ICRA), 2012 IEEE International Conference on, IEEE, 2012, pp. 842–849.

[46] Y. Zhu, W. Chen, G. Guo, Evaluating spatiotemporal interest point features for depth-based action recognition, Image and Vision Computing 32 (8) (2014) 453–464.

[47] D. R. Faria, C. Premebida, U. Nunes, A probabilistic approach for human everyday activities recognition using body motion from rgb-d images, in: Robot and Human Interactive Communication, 2014 RO-MAN: The 23rd IEEE International Symposium on, IEEE, 2014, pp. 732–737.

[48] J. Shan, S. Akella, 3d human action segmentation and recognition using pose kinetic energy, in: Advanced Robotics and its Social Impacts (ARSO), 2014 IEEE Workshop on, IEEE, 2014, pp. 69–75.

[49] S. Gaglio, G. L. Re, M. Morana, Human activity recognition process using 3-d posture data, IEEE Transactions on Human-Machine Systems 45 (5) (2015) 586–597.

[50] G. I. Parisi, C. Weber, S. Wermter, Self-organizing neural integration of pose-motion features for human action recognition, Frontiers in neurorobotics 9 (2015) 3.

[51] E. Cippitelli, S. Gasparrini, E. Gambi, S. Spinsante, A human activity recognition system using skeleton data from rgbd sensors, Computational intelligence and neuroscience 2016 (2016) 21.

[52] N. Hu, G. Englebienne, Z. Lou, B. Kröse, Learning latent structure for activity recognition, in: Robotics and Automation (ICRA), 2014 IEEE International Conference on, IEEE, 2014, pp. 1048–1053.

[53] A. Taha, H. H. Zayed, M. Khalifa, E.-S. M. El-Horbaty, Skeleton-based human activity recognition for video surveillance, International Journal of Scientific & Engineering Research 6 (1).

[54] H. S. Koppula, A. Saxena, Anticipating human activities using object affordances for reactive robotic response, IEEE transactions on pattern analysis and machine intelligence 38 (1) (2016) 14–29.

[55] Z. Liu, C. Zhang, Y. Tian, 3d-based deep convolutional neural network for action recognition with depth sequences, Image and Vision Computing 55 (2016) 93–100.

[56] Y. Yang, C. Deng, D. Tao, S. Zhang, W. Liu, X. Gao, Latent max-margin multitask learning with skelets for 3-d action recognition, IEEE transactions on cybernetics.

[57] M. Antunes, D. Aouada, B. Ottersten, A revisit to human action recognition from depth sequences: Guided svm-sampling for joint selection, in: Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on, IEEE, 2016, pp. 1–8.

[58] C. Ellis, S. Z. Masood, M. F. Tappen, J. J. LaViola, R. Sukthankar, Exploring the trade-off between accuracy and observational latency in action recognition, International Journal of Computer Vision 101 (3) (2013) 420–436.

[59] X. Jiang, F. Zhong, Q. Peng, X. Qin, Robust action recognition based on a hierarchical model, in: Cyberworlds (CW), 2013 International Conference on, IEEE, 2013, pp. 191–198.

[60] T. Kerola, N. Inoue, K. Shinoda, Spectral graph skeletons for 3d action recognition, in: Asian Conference on Computer Vision, Springer, 2014, pp. 417–432.

[61] J. Beh, D. K. Han, R. Durasiwami, H. Ko, Hidden markov model on a unit hypersphere space for gesture trajectory recognition, Pattern Recognition Letters 36 (2014) 144–153.

[62] W. Ding, K. Liu, F. Cheng, J. Zhang, Stfc: spatio-temporal feature chain for skeleton-based human action recognition, Journal of Visual Communication and Image Representation 26 (2015) 329–337.

[63] G. Lu, Y. Zhou, X. Li, M. Kudo, Efficient action recognition via local position offset of 3d skeletal body joints, Multimedia Tools and Applications 75 (6) (2016) 3479–3494.

[64] S. Gasparrini, E. Cippitelli, E. Gambi, S. Spinsante, J. Wåhslén, I. Orhan, T. Lindh, Proposal and experimental evaluation of fall detection solution based on wearable and depth data fusion, in: ICT Innovations 2015, Springer, 2016, pp. 99–108.