Skeleton-based Human Action Recognition
A Learning Method based on Active Joints

Ahmad K. N. Tehrani, Maryam Asadi Aghbolaghi and Shohreh Kasaei
Department of Computer Engineering, Sharif University of Technology, Tehran, Iran
{*khataminejad, masadia*}*@ce.sharif.edu, skasaei@sharif.edu*

Keywords: Human Action Recognition, Active Joints, Hidden Marcov Model (HMM), Skeletal Human Body Model.

Abstract: A novel method for human action recognition from the sequence of skeletal data is presented in this paper. The proposed method is based on the idea that some of body joints are inactive and do not have any physical meaning during performing an action. In other words, regardless of the subjects that perform an action, for each action only a certain set of joints are meaningfully involved. Consequently, extracting features from inactive joints is a time-consuming task. To cope with this problem, in this paper, only the dynamic of active joints is modeled. To consider the local temporal information, a sliding window is used to divide the trajectory of active joints into some consecutive windows. Feature extraction is then applied on all windows of active joints' trajectories and then by using the K-means clustering all features are quantized. Since each action has its own active joints, in this paper one-vs-all classification strategy is exploited. Finally, to take into account the global motion information, the consecutive quantized features of the samples of an action are fed into the *hidden Markov model* (HMM) of that action. The experimental results show that using active joints can get 96% of maximum reachable accuracy from using all joints.

1 INTRODUCTION

Human action recognition is one of the most widely studied research topics in computer vision (because it has been massively applied in many real-world applications like health care systems, video analysis, and so forth). In the past decades, research of *human action recognition* has mainly focused on recognizing actions from videos captured by traditional visible light cameras. In the recent years, it has continued to be a hot area of research in computer vision thanks to the emergence of low-cost depth sensors like Kinect. It has some advantages over the visible light cameras (Xia and Aggarwal, 2013). First, the 3D structure information of the environment can be obtained, which provides more discriminative knowledge. Next, the depth data is independent of environment brightness and it can capture depth images even in total darkness. Moreover, in (Shotton et al., 2013), a real-time method is proposed which estimated human joints 3D positions based on an algorithm that extracts body parts from the depth data.

Almost all skeletal-based approaches extract features from all body joints. It is noteworthy that most of the time, some joints are not involved; for instance, neither are lower body joints effective during hand-waving action nor is hand's motion effective during the "forward-kick" action. Therefore, for each action, joints can be categorized into two groups of active and inactive joints. Active joints for one class of action are the joints which are dynamic during performing that action and inactive joints are the ones that do not have any significant motions. Regarding this fact, a novel method is proposed in this paper which uses only active joints of each action for feature extraction and classification of human actions (Fig. 1).

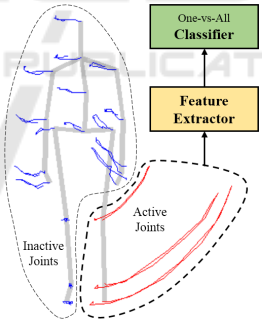


Figure 1: General idea of proposed method. The trajectories show the position of each joint during the action.

It seems that humans use the idea of weighting most important joints to recognize each action. For example, during viewing an unknown action which

K. N. Tehrani A., Asadi Aghbolaghi M. and Kasaei S.
Skeleton-based Human Action Recognition - A Learning Method based on Active Joints.
DOI: 10.5220/0006134903030310
In *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2017)*, pages 303-310
ISBN: 978-989-758-226-4

has some effective motions on hands and no effective movements on feet, humankind ensures that performed action does not belong to foot-based action categories such as "ball-shooting". In addition, if one knows that a certain action has been performed by moving hands, for recognizing the performed action, the attention should be focused on the kind of hands' motion rather than feet's motion, which have been inactive in this case. This observation intuitively shows that the proposed method can work thoroughly.

Action recognition needs two main parts: feature extraction and learning method. There are different types of feature extraction methods some of which are presented in Section 2. Moreover, to learn extracted feature vector, two main methods are used: first, using a multi-class classifier to learn all of feature vectors and the other, using one-vs-all classifier for each class beside a unit which selects the best class based on the results of classifiers.

In this paper, the main goal is representing a learning method which is able to manage different kinds of active joints per action. To achieve this, based on what was stated before about the behavior of humankind in recognizing actions, two different approaches are presented, one for training and one for testing. In the training phase, firstly, by using a proposed unit, the most active joints for each class of actions are selected. Then, after extracting features from each window of active joints' trajectory and constructing the full feature vector of each window, all feature vectors are quantized by using a clustering algorithm. Finally, a one-vs-all classifier (HMM) is learned for each class of actions by using the samples of that class. In the testing phase, firstly, active joints of each sample are selected. Then, the full feature vector is converted to a quantized sequence (by using the center of each cluster), It is then fed into the classifier of probable action (explained in Section 3). Finally, a decision is made from yielded certainty factor of probable classes.

The main contributions consist of four parts: First, the idea of using few active joints to recognize the action instead of using all joints. Second, an approach has been proposed to select active joints from a sample of an action. Third, a method to learn the training data has been presented based on differences between active joints involved in actions of different classes. Fourth, an approach has been proposed based on the effects of active joints on an action class.

2 RELATED WORK

The recent advent of low-cost and easy-operation depth sensors like Kinect have received a great deal of attention from researchers to reconsider problems such as activity recognition using depth images instead of color images. Generally, recent approaches in 3D action recognition can be divided into two groups based on the input data of depth data or skeleton information. 3D skeletal information is at a higher level of semantic rather than depth information which has less data to be processed. As a result, the skeleton-based descriptor is more discriminative for presenting human actions.

The 3D skeletal information can be obtained in different ways. The most accurate ones are provided by motion capture (F. De la Torre and Macey., 2009) which are typically captured using optical sensing of markers placed on specific positions of human body. Although these data are more reliable and less noisy than the other sources of skeletal data, it is so difficult to produce such data. The 3D joints locations can be also estimated accurately from depth map by (Shotton et al., 2013) in real-time. This algorithm brings many benefits to numerous tasks in computer vision especially in action recognition. Many approaches have been proposed to recognize human actions using 3D joints locations. There are some kinds of features that can be extracted from joints; i.e., raw-based, displacement-based, and orientation-based features.

In (Hussein et al., 2013), the statistical *covariance of raw 3D joints positions* (Cov3DJ) is used as the feature. Some other methods like (Wei et al., 2013) have formed the trajectories using the raw positions of joints and then extract features from the trajectories. Temporal information can be processed in two ways for trajectory-based methods. Some approaches exploit histogram of spatio-temporal extracted features and some other approaches use temporal analysis tools like HMM, *self-similarity matrix* (SSM), and *dynamic time warping* (DTW) in (Xia et al., 2012), (Junejo et al., 2011), and (Vemulapalli et al., 2014), respectively. In (Xia et al., 2012), HOJ3D is proposed as a descriptor for action recognition from skeletal data. To make this method invariant to rotation, a special spherical coordinate is defined based on 3D location of joints, such that the hip joint is used as the center of coordinate and the vector from left hip to right hip is defined as the horizontal axis.

Displacement-based approaches usually use two kinds of displacements, spatial and temporal. Spatial features like (Yang and Tian, 2012; Luo et al., 2013) are the ones which are extracted from all joints locations in just one frame like joints pairwise distances and the temporal features like (Yang and Tian, 2012) are the ones which measure the movement of one joint during the time, for instance, joints velocity and acceleration.

Oriented-based methods exploited the joints orientation as their features. These features can also be spatial or temporal. Spatial oriented-based features such as (Gu et al., 2012) are the orientation of displacement vectors of a pair of human skeletal joints in one frame and the temporal features such as (Boubou and Suzuki, 2015) are the ones that compute difference between orientations of each joint during the time. In (Eweiwi et al., 2014), three kinds of features are extracted from skeletal data of each joint in all frames: 3D histograms of joint location, 2D histograms of velocity vectors, and 2D histogram of the cross product of location vector and velocity vector.

3 PROPOSED METHOD

Each recognition problem needs to use two main sections: feature extraction and learning. In this paper, A novel way of learning is designed which can discriminate between different joints with different amount of activity.

When an action is performed, to recognize an action, one focuses on joints which have the most effective motions. For instance, when someone drinks water, if you concentrate on the feet's motion, you may not find out which action is performed. Therefore, the only way to recognize each action is to focus on its effective joints (called *active joint* in this paper). In addition, to recognize each action, one evaluates how active joints move during the action period. Ultimately, when an action is performed, one can recognize it by answering two key questions:

1. Which joints are involved in that action?
2. How does each joint move during the action time?

Answering these two questions can lead to recognizing each action.

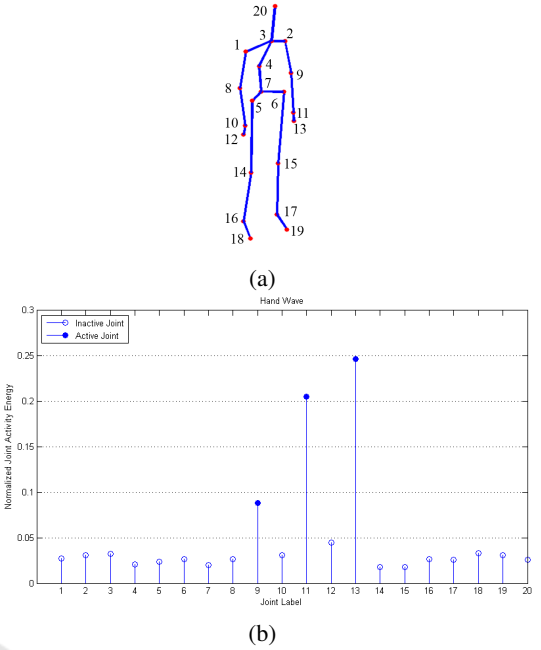
The expression *active joint* must be clarified before starting to explain the proposed method. The active joints are the joints that have the most effective movements during the action. To measure the amount of activity of each joint, an energy function is defined on each joint, given by

$$e_i = \sum_{t \in T} \| P_i(t) - P_i(t-1) \|_2 \qquad (1)$$

where e_i is the energy of joint i which $P_i(t)$ relates to its position at frame t of total action time (T).

To make this energy function comparable with other samples, a normalization factor is applied on it. Joints' energies are normalized by

$$E_i = \frac{e_i}{\sum_j e_j} \qquad (2)$$



(a)

(b)

Figure 2: (a) Joints number in skeleton model. (b) Effect of applying threshold ($\alpha = 0$) on normalized joint energy for "hand-waving" action.

where E_i is the normalized energy of joint i which has been scaled to the summation of all joints (j) energy.

Normalized joint energy for the action "hand-waving" is illustrated in Fig. 2(b) (based on skeleton model in Fig. 2(a)). As is expected, the amount of joint energy for joints 9, 11, and 13 (related to the left hand) is greater than other joints' energy.

Proposed learning algorithm is inspired from the idea introduced above. To treat in this manner (magnifying the effect of active joints on recognition), two main sections are required: one for training and another for testing. In the following, the functionality of each part is explained.

3.1 Training Phase

This part is designed based on using active joints. The main idea of this part is illustrated in Fig. 3. The block diagram of training phase illustrates that the proposed method firstly finds the set of active joints (by the "Active Joint Selector" unit) for each class of training data. Then, for each training sample, features are extracted from windows which are sliding on the trajectory of active joints. After extracting features from windows of trajectory of active joints, they are concatenated together (for inactive joints, a fixed set of numbers is placed) to construct a main feature vector for each window. By applying a vector quantization method (K-means) on all main feature vec-

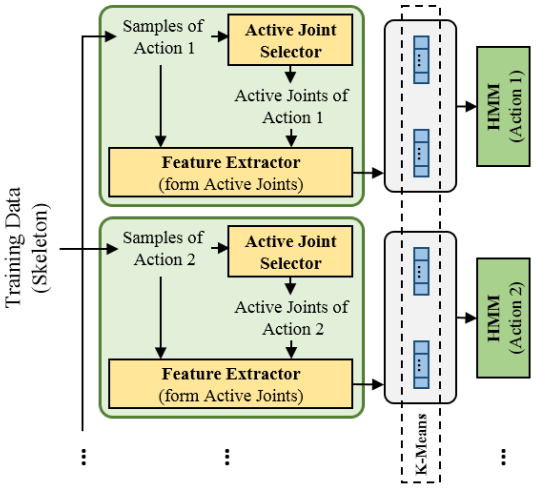


Figure 3: Block diagram of training phase.

tors of samples of all classes, they will be converted to a quantized sequence. Finally, the quantized sequence of samples of each class are used for learning the HMM (as a one-vs-all classifier) for that class. Using HMM in proposed method causes to manage effect of time that is eliminated from feature vectors by using the feature extraction method which are saving spatial information.

Fig. 3 shows that working with this kind of learning method needs some subsequent processes. These will be explained in the following subsections.

3.1.1 Active Joint Selector

This part of the proposed method must select the most active joints. The amount of joint activity is represented by the normalized joint energy which was stated before.

To select some of the joints as active joints from the normalized energy of each sample, an adaptive threshold must be applied on them. The adaptive threshold is given by

$$thr = mean\{E_{1,\dots,I}\} + \alpha * std\{E_{1,\dots,I}\} \qquad (3)$$

where $E_{1,\dots,I}$ and α are the normalized energy of skeleton model of each action sample and the alignment factor (which can change the accuracy of selecting active joints), respectively.

Therefore, by applying this threshold on the joint energy of skeleton model, active joints can be selected. Result of selecting active joints for instances of the action hand-waving is illustrated in Fig. 2(b).

What was stated before was a method to select active joints of each sample. However, in the training phase, selecting active joints must be more general than each sample, due to the noise which has been distributed on each sample joint energy. To select active joints of each class, declared threshold must be applied on the average of normalized energy of all instances of each class.

3.1.2 Feature Extraction

The main idea of this paper is its learning method. However, to test the proposed learning method, an approach for feature extraction is required. Feature extraction method which is used during testing this learning method is stated in the following.

To extract features, firstly, a sliding window affects the trajectory of selected active joints. Then, a feature extraction method is applied on each window of trajectories. The important and remarkable point of this type of feature extraction is that extracted features describe local characteristics of each joint's trajectory. In other words, the proposed method attempts to describe each trajectory by a set of local features instead of a global extracted feature vector which describes the whole trajectory. To describe local characteristics of trajectories, the feature extraction method which has been applied on each slided window of trajectories, consists of some parts:

1. Multi-level wavelet decomposition: This transform (Primer et al., 1998) can break each signal into two parts: approximation and detail. Coefficients of this transform function which are applied on the window of joint trajectory are used as part of feature vector.
2. Discrete cosine Transform: This transform can convert a finite signal into sum of cosine functions with different frequencies.
3. Fast Fourier transform: The magnitude and phase of each window of trajectory of the joint is used as part of feature vector.
4. Displacement of joint w.r.t. joint "torso" for each frame of applied window on the joint trajectory (Rahmani et al., 2014).
5. Cov3DJ: This feature vector was introduced in (Hussein et al., 2013). In this paper, Cov3DJ is modified by using just one joint instead of all.
6. Displacement of joint at the first frame of window w.r.t. the next frame.
7. Difference between maximum and minimum of joint trajectory located in each window (Rahmani et al., 2014).

After extracting feature vector from each window of active joints, now it is time to generate the main and large feature vector which describes a period of time

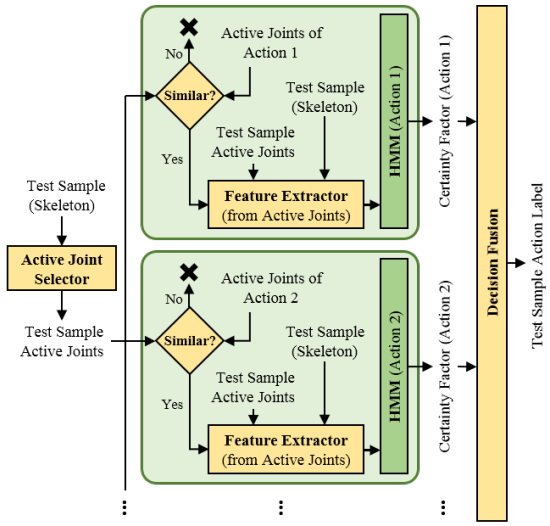


Figure 4: Block diagram of test phase.

of action instance (related to each widow). To construct the main feature vector of each period of time, feature vector of each active joint and a set of fixed numbers for each inactive joint must be concatenated to each other.

3.2 Test Phase

As asserted before, this method is trying to add the concept of active joints in each main part. This concept also shows its effect on the test phase as follows.

The main idea of the test phase is illustrated in Fig. 4. According to that block diagram, for each test instance, firstly, active joints are selected by the unit which has been previously defined (Active Joint Selector). Next, the similarity between test active joints and active joints of each class are evaluated. In this situation, two cases may occur. If they were not acceptably similar (like active joints between "hand-waving" and "ball-shooting"), the certainty of not belonging to that class is in a high level. In contrast, If their similarity is acceptable, initially features are extracted from active joints and then they will be concatenated to each other (instead of inactive joints, a fixed set of numbers is placed). Next, by comparing extracted feature vectors with center of each cluster (obtained by K-means) quantized sequence goes to be evaluated with pre-learned HMM. The output of HMM will be a certainty factor which represents the similarity of test data and The HMM's class.

The similarity unit works based on comparing the most active joints of test action and active joints of each class. If they have more mutual joints than a fixed percentage of active joints of the class being compared, their similarity is considered acceptable.

Figure 5: Input of "Decision Fusion".

Eventually, the "Decision Fusion" unit is required to choose the best label for test data. Input of this unit is similar to the one illustrated in Fig. 5. According to the functionality of this unit, an action label must be selected from relevant classes by selecting the class with maximum certainty factor.

4 EXPERIMENTAL RESULTS

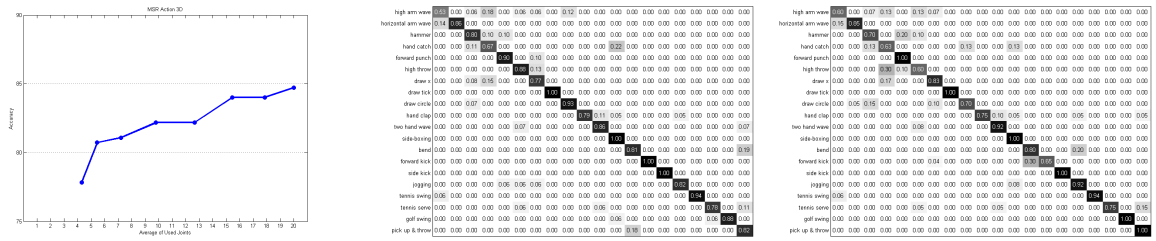
To assess the proposed method, two datasets are used: MSR Action 3D (Li et al., 2010) and UTKinect (Xia et al., 2012). In the following, the performance analysis on these datasets is given based on the parameters which have been selected experimentally.

4.1 MSR Action 3D

MSR action 3D is a gaming action dataset with depth sequences. It includes 20 actions: *high arm wave*, *horizontal arm wave*, *hammer*, *hand catch*, *forward punch*, *high throw*, *draw x*, *draw tick*, *draw circle*, *hand clap*, *two hand wave*, *side boxing*, *bend*, *forward kick*, *side kick*, *jogging*, *tennis swing*, *tennis serve*, *golf swing*, and *pick up and throw*. Each action is performed 2 or 3 times by 10 subjects. The frame rate is 15 fps and its resolution is 320×240.

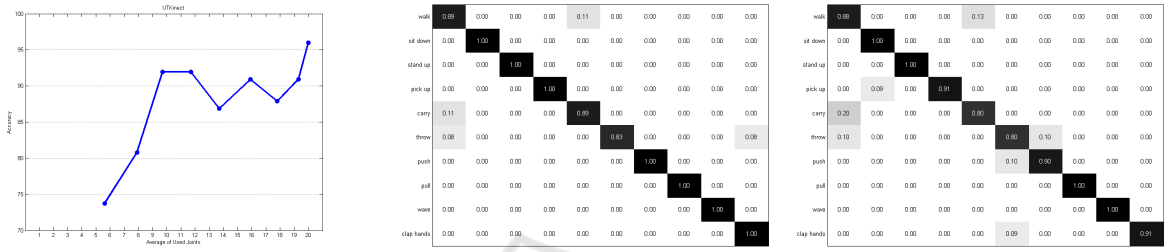
To test the proposed method on MSR action 3D, the window size and the overlap between two windows for feature extraction have been set to 8 and 7 frames, respectively. Then, all feature vectors have been quantized into 400 clusters and finally by setting the number of HMM state to 15, training and testing phases are conducted. Train and test are done by changing factor α which has been previously stated (Equation 3). By using a fixed α, "Active Joint Selector" can select active joints of each class of action and finally by averaging over the number of active joints for each action, the average of active joints for that experiment (including both train and test) is calculated.

Test accuracy results for different averages of active joints are given in Table 1. Obviously, using the active joints instead of all joints can speed up the feature extraction step. As is clear in Table 1, by using about 9.8 of all joints, obtaining an accuracy of 97% of using all joints is accessible. The relation between



(a) (b) (c)

Figure 6: MSR action 3D results. (a) Effect of using different average amount of active joints at each experiment. (b) Confusion matrix of all joints. (c) Confusion matrix of 9.8 of joints on average.



(a) (b) (c)

Figure 7: UTKinect results. (a) Effect of using different average amount of active joints at each experiment. (b) Confusion matrix of all joints. (c) Confusion matrix of 9.7 of joints on average.

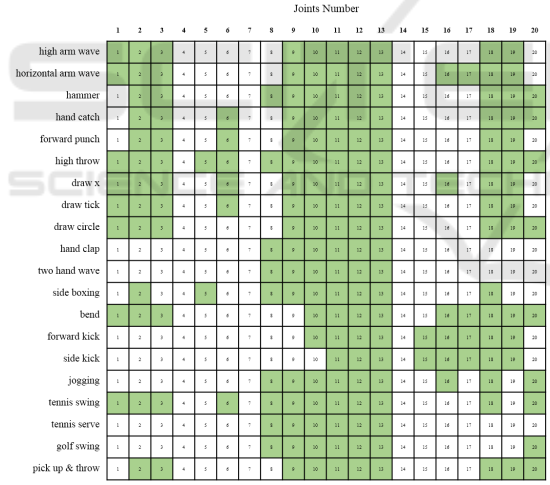


Figure 8: Selected active joints of each class for $\alpha = 0.372$ on MSR action 3D dataset.

accuracies when using all joints and active joints is illustrated in Fig. 6(a).

In order to get the average of active joints to 9.8, α is assigned to 0.372 ($\alpha = 0.372$). By using this amount of α, the list of active joints per action is illustrated in Fig. 8 based on skeleton model of body which is illustrated in 2(a). Confusion matrices for the case "all joints" and "average of active joints = 9.8" are given in 6(b) and 6(c), respectively. From the reported confusion matrix for active joints and the list of active joints in case $\alpha = 0.372$, this conclusion is yielded that there is no error and mis-recognizing item when active joints are different. Results of the proposed and previous methods are given in Table 3.

The result on MSR action 3D dataset is lower than some other methods. This is due to the problem that a set of actions in this dataset have the same active joints. Most of the misclassified actions are in this set. Thus, it can be analyzed that using better feature extraction methods will solve this problem.

4.2 UTKinect

UTKinect is another action dataset which contains both RGB and depth video sequences. The depth map frame rate and the resolution are 30 fps and 320×240, respectively. Also, the resolution of RGB sequences is 640×480. This dataset includes 10 actions: *walk*, *sit down*, *stand up*, *pick up*, *carry*, *throw*, *push*, *pull*, *wave*, and *clap hands*. Actions of this dataset have been performed twice by 10 subjects. The actions in this dataset cover the movements of hands, arms, legs, and upper torso.

To test the proposed method based on active joints on UTKinect, the window size and the overlap between windows in order to extract feature, are set to 4 and 3 frames, respectively. The number of clusters to be used for K-means for quantization of extracted feature vectors, is set to 120. To train the HMM of each class, 5 states have been considered.

The test results for different averages of active joints are given in Table 2 for UTKinct. As it is stated

Table 1: MSR action 3D results.

Used Active Joints	4.3	5.45	7.2	9.8	12.65	15.45	17.85	20
Accuracy %	77.82	80.73	81.09	82.18	82.18	84.00	84.00	84.72
Scaled Accuracy* %	91.85	95.29	95.71	97.00	97.00	99.15	99.15	Ref.
Speed up** %	365	267	178	104	58	29	12	Ref.

* The accuracies have been scaled w.r.t. the accuracy when all joints are used.
** The percentage of speeding up, when fewer joints are used instead of all.

Table 2: UTKinect results.

Used Active Joints	5.6	7.9	9.7	11.7	13.7	15.9	17.8	19.3	20
Accuracy %	73.74	80.81	91.92	91.92	86.87	90.91	87.88	90.91	95.96
Scaled Accuracy* %	76.84	84.21	95.79	90.53	94.73	91.58	94.77	94.74	Ref.
Speed up** %	257	153	106	71	46	26	12	4	Ref.

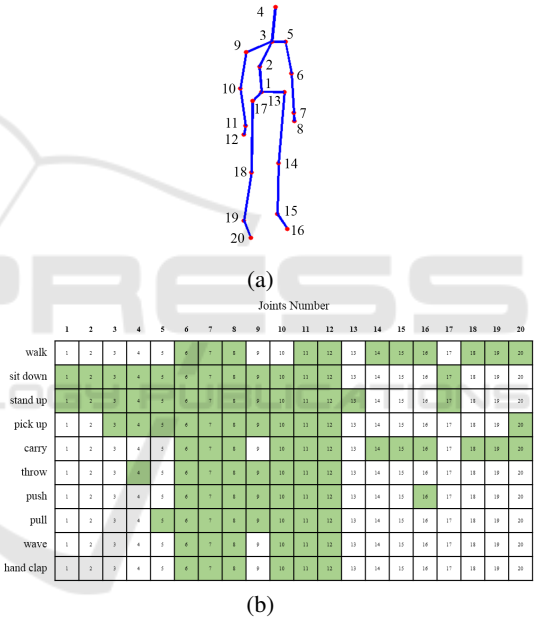
* Accuracies have been scaled w.r.t. when all joints are used.
** Percentage of speeding up, when fewer joints are used instead of all.

Table 3: Performance of proposed method on MSR action 3D dataset, compared to previous approaches.

Method	Accuracy %
(Li et al., 2010)	74.70
(Xia et al., 2012)	79.00
(Oreifej and Liu, 2013)	88.89
(Yang and Tian, 2014)	**93.09**
Proposed (Using all joints)	84.72
Proposed (Using 9.8 of joints)	82.18

for MSR action 3D, using active joints speeds up feature extraction. Table 2 shows that using 9.7 joints on average obtains 96% of maximum available accuracy when all joints are used. In this case, features can be extracted about 2 times faster than the case for which all joints are used. The relation between accuracies in using all joints and using active joints (Fig. 7(a)) shows that by increasing the number of active joints (instead of all joints), the accuracy decreases rather than the case that 9.7 of joints are used.

If the average of active joints gets to 9.7, α must be set to 0.3 ($\alpha = 0.3$). Active joints of UTKinect dataset in this case are illustrated in 9(b) (Joints number is based on Fig 9(a)). The confusion matrix for the case "all joints" and "average of active joints = 9.7" are given in 6(b) and 6(c), respectively. Results of the proposed and previous methods are given in Table 4.



(a)

(b)

Figure 9: (a) Joints number of skeleton model in UTKinect dataset. (b) Selected active joints of each class for $\alpha = 0.3$ on UTKinect dataset.

Table 4: Performance of proposed method on UTKinect dataset, compared to previous approaches.

Method	Accuracy %
(Xia et al., 2012)	90.92
(Devanne et al., 2013)	91.5
(Liu et al., 2015)	95.00
(Vemulapalli et al., 2014)	**97.08**
Proposed (Using all joints)	95.96
Proposed (Using 9.7 of joints)	91.92

5 CONCLUSION

A novel method for human action recognition from the sequences of skeletal human body data was presented. Using the skeletal data make this algorithm applicable to the "motion capture" problem. The proposed approach was founded upon the idea that given an action, there are just some active joints. Thus, existence of active joints is sufficient and there is no dependence on the existence of inactive joints of human

body, which makes this algorithm more flexible than the other ones. This characteristic and using skeletal data, made this algorithm to be used in the case for which the information of some inactive joints are missing. The local temporal features were extracted using overlapped sliding windows over the trajectory of each joint, and the global temporal information was taken into account using the HMM classifier. Also, there is a rich possibility for extensions. In this paper, there is no contribution to feature extraction, and this belief exists that by using more discriminative features, the final accuracy of the method can be improved. Thus, as a future work, the state-of-the-art feature extraction methods can be used. Using more powerful quantization method instead of K-means can also improve the results.

REFERENCES

Boubou, S. and Suzuki, E. (2015). Classifying actions based on histogram of oriented velocity vectors. *Journal of Intelligent Information Systems*, 44(1):49–65.

Devanne, M., Wannous, H., Berretti, S., Pala, P., Daoudi, M., and Del Bimbo, A. (2013). Space-time pose representation for 3d human action recognition. In *International Conference on Image Analysis and Processing*, pages 456–464. Springer.

Eweiwi, A., Cheema, M. S., Bauckhage, C., and Gall, J. (2014). Efficient pose-based action recognition. In *Asian Conference on Computer Vision*, pages 428–443. Springer.

F. De la Torre, J. Hodgins, J. M. S. V. R. F. and Macey., J. (2009). Tech. report cmu-ri-tr-08-22. Technical report, Robotics Institute, Carnegie Mellon University.

Gu, Y., Do, H., Ou, Y., and Sheng, W. (2012). Human gesture recognition through a kinect sensor. In *Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on*, pages 1379–1384. IEEE.

Hussein, M. E., Torki, M., Gowayyed, M. A., and El-Saban, M. (2013). Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. In *IJCAI*, volume 13, pages 2466–2472.

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

Li, W., Zhang, Z., and Liu, Z. (2010). Action recognition based on a bag of 3d points. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*, pages 9–14. IEEE.

Liu, Z., Feng, X., and Tian, Y. (2015). An effective view and time-invariant action recognition method based on depth videos. In *2015 Visual Communications and Image Processing (VCIP)*, pages 1–4. IEEE.

Luo, J., Wang, W., and Qi, H. (2013). Group sparsity and geometry constrained dictionary learning for action recognition from depth maps. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1809–1816.

Oreifej, O. and Liu, Z. (2013). Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 716–723.

Primer, A., Burrus, C. S., and Gopinath, R. A. (1998). Introduction to wavelets and wavelet transforms.

Rahmani, H., Mahmood, A., Huynh, D. Q., and Mian, A. (2014). Real time action recognition using histograms of depth gradients and random decision forests. In *IEEE Winter Conference on Applications of Computer Vision*, pages 626–633. IEEE.

Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., and Moore, R. (2013). Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124.

Vemulapalli, R., Arrate, F., and Chellappa, R. (2014). 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*, pages 588–595.

Wei, P., Zheng, N., Zhao, Y., and Zhu, S.-C. (2013). Concurrent action detection with structural prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3136–3143.

Xia, L. and Aggarwal, J. (2013). Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2834–2841.

Xia, L., Chen, C.-C., and Aggarwal, J. (2012). View invariant human action recognition using histograms of 3d joints. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 20–27. IEEE.

Yang, X. and Tian, Y. (2014). Super normal vector for activity recognition using depth sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 804–811.

Yang, X. and Tian, Y. L. (2012). Eigenjoints-based action recognition using naive-bayes-nearest-neighbor. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 14–19. IEEE.