# 3D Human Activity Recognition Using Skeletal Data from RGBD Sensors

Jiaxu Ling, Lihua Tian, and Chen Li(✉)

School of Software Engineering, Xi'an Jiaotong University, Xi'an, China
lingjiaxv@stu.xjtu.edu.cn, {lhtian,cclidd}@xjtu.edu.cn

**Abstract.** In this paper, a new effective method was proposed to recognize human actions based on RGBD data sensed by a depth camera, namely Microsoft Kinect. Skeleton data extracted from depth images was utilized to generate 10 direction features which represent specific body parts and 11 position features which represent specific human joints. The fusion features composed of both was used to represent a human posture. An algorithm based on the difference level of adjacent postures was presented to select the key postures from an action. Finally, the action features, composed of the key postures' features, were classified and recognized by a multiclass Support Vector Machine. Our major contributions are proposing a new framework to recognize the users' actions and a simple and effective method to select the key postures. The recognition results in the KARD dataset and the Florence 3D Action dataset show that our approach significantly outperforms the compared methods.

## 1 Introduction

In recent years, the computer vision technology has advanced rapidly with the introduction of consumer depth cameras with real-time capabilities, such as Microsoft Kinect. These new devices have stimulated the development of various promising applications, including human posture reconstruction and estimation [1], scene flow estimation [2], hand gesture recognition [3] and face super-resolution [4]. Human action recognition has also become a trending research topic since it may be applied in many areas such as gaming, human-computer interaction and medical assistance.

It is a good choice to design action recognition systems using depth maps provided by depth sensors, like Microsoft Kinect or other similar devices because they can provide human body contour, which simplifies the task of human body detection and segmentation [5]. In addition, a simple and effective representation of the human body by skeletal joints data extracted from the depth frames has a broad application prospect [6]. However, the existing methods which aim to recognize the human actions still have limitations in practice. The major challenge comes from noise, missing data, the similarity of human actions and the variety of the same action.

Early techniques in human action recognition mainly deal with processing the color images provided by RGB cameras. In [7], the silhouettes extracted from RGB data are used to represent the human postures and are input into a framework based on HMM. The silhouettes and discrete HMMs are also used in [8], where Fourier Analysis is applied to describe the human silhouettes and SVMs [9] is used to classify them into


G. Bebis et al. (Eds.): ISVC 2016, Part II, LNCS 10073, pp. 133–142, 2016.
DOI: 10.1007/978-3-319-50832-0_14

different postures. There are some deficiencies with the RGB data-based methods, such as the complexity in the processing chain and limitation in real-time use. With the introduction of consumer depth cameras, many methods based on RGBD data are proposed. These methods can be divided into three categories: (1) methods using depth data; (2) methods using skeleton data; (3) methods using hybrid data.

The methods simply using depth data. The approach in [10] projects depth maps onto three orthogonal planes and accumulates global actions through entire video sequences to generate the Depth Motion Maps (DMM). Histograms of Oriented Gradients are then computed from DMM as the representation of an action video. The HON4D descriptor [11] is based on the orientations of normal surfaces in 4D.

The methods simply using skeleton data. The APJ3D representation [12] is composed of the relative positions and local spherical angles, which are computed from a subset of 15 skeleton joints. After selecting the key postures, the action is partitioned by a reviewed Fourier Temporal Pyramid [13] and then classified by random forests.

The methods using both depth and skeleton data. Althloothi et al. [14] proposed a method by which the data is fused with the Multiple Kernel Learning (MKL) technique at the kernel level, instead of the feature level. In [15], actions are characterized using pairwise affinity measures between joint angle features and histogram of oriented gradients computed on depth maps.

The approach proposed in this paper mainly uses the skeleton data extracted by Kinect. The main process can be divided into three steps. Firstly, the skeleton data extracted from depth images is utilized to generate 10 direction features which represent specific body parts and 11 position features which represent specific human joints, the fusion features composed of which were used to represent a human posture. Secondly, an algorithm based on the difference level of adjacent postures is presented to extract the key postures from an action. Finally, the action features, composed of key postures' features, as input for a multiclass Support Vector Machine.

## 2 Action Recognition Algorithm

The proposed algorithm in this paper uses the skeleton data extracted by Kinect. The main process can be divided into posture feature extraction, the key posture selection and classification. The major point is that an action is composed of a set of postures in essence, so that we can select some key postures to represent an action, and then classify the actions with a multiclass SVM. Figure 1 is the flow chart of the proposed algorithm and the following is a simple description of the three main steps:

(1) Posture feature extraction: The direction vector composed of the main human bodies' direction in physical space can represent a posture and the position vector composed of the main human joints' coordinates in physical space can also represent a posture. So we will use the fusion features of both to represent a posture.
(2) Key posture selection: Each posture sequence of an action has different length in all probability and the next step requires the input of the same length, so we need to select the k most important postures to represent an action.

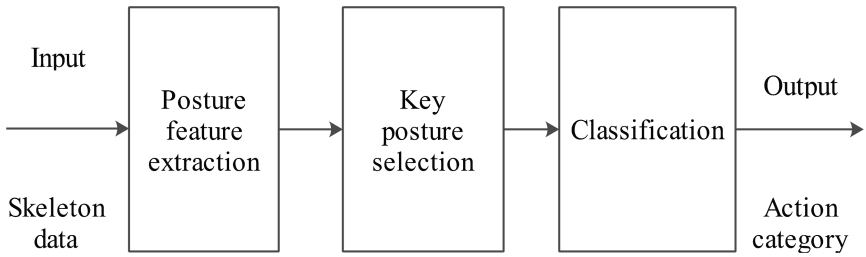


**Fig. 1.** The flow chart of proposed algorithm. The input is the skeleton data extracted by Kinect. After the three main steps: posture feature extraction, key posture selection and classification, the system can recognize the actions.

(3) Posture classification: We first combine the k key postures we get from the previous step into action features in linear and then classify them with a multiclass Support Vector Machine.

### 2.1 Posture Feature Extraction

The Kinect platform can recognize each human body in the acquired RGBD datastream and then output a wireframe skeleton at a rate of 30 fps. Each skeleton includes 20 joints of human body parts, such as the hip, the upper arms and the head, etc. The position of the skeleton joints is provided as (x, y, z) coordinates in an absolute reference system that places the Kinect device at the origin with the positive z-axis extending in the direction in which the device is pointed, the positive y-axis extending upward, and the positive x-axis extending to the left.

There are many methods to figure out a variety of features to represent a posture based on skeleton data. In this paper, two kinds of effective feature extraction methods are used. One is the direction vector using the directions of the main human body parts, and the other is the position vector using the coordinates of the main human joints. The two methods are introduced in [16, 17].

The direction vector, as shown in Fig. 2(a), selects 10 main human body parts (upper arms, forearms, thighs, crura, head and spine). The computation formula of direction vector is as follows:

$$D_i = \frac{j_m - j_n}{||j_m - j_n||}, j_m, j_n \in J \tag{1}$$

Where, j is the coordinates of a human joint, provided as {x, y, z}, and J is a collection of all the human joints. D is a direction vector of a body part. The proposed algorithm selects 10 main human body parts, which is shown in Table 1.

Position vector, as shown in Fig. 2(b), selects 11 joints of main human body parts (head, neck, torso, elbows, hands, knees and feet). The position information of joints cannot be regarded as the features directly, so we need to normalize it. Let $d_i$ be the vector containing the 3-D normalized coordinates of the joint $j_i$. J is a collection of the

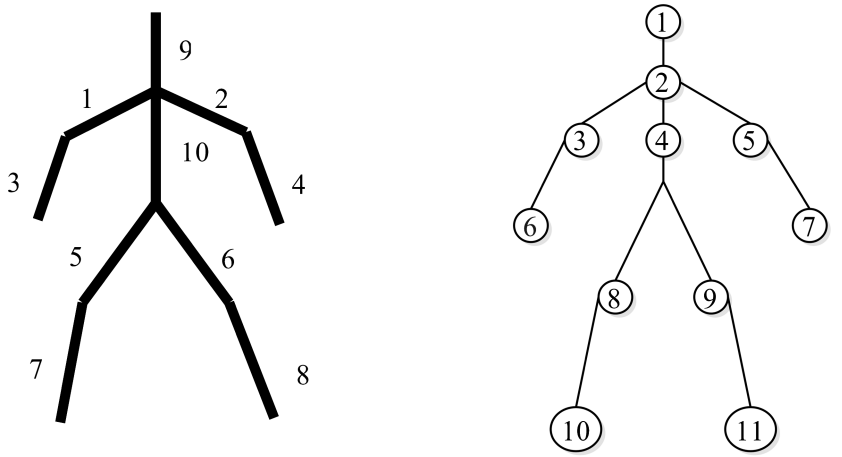


**Fig. 2.** (a) Ten selected human body parts: upper arms, forearms, thighs, crura, head and spine. (b) Eleven selected joins: head, neck, torso, elbows, hands, knees and feet.

**Table 1.** List of direction features

| D | $j_m$ | $j_n$ | Description |
|---|---|---|---|
| 1 | Left elbow | Left shoulder | Left upper arm |
| 2 | Right elbow | Right shoulder | Right upper arm |
| 3 | Left wrist | Left elbow | Left forearm |
| 4 | Right wrist | Right elbow | Right forearm |
| 5 | Left knee | Left hip | Left thigh |
| 6 | Right knee | Right hip | Right thigh |
| 7 | Left ankle | Left knee | Left crus |
| 8 | Right ankle | Right knee | Right crus |
| 9 | Hip center | Shoulder center | Spine |
| 10 | Shoulder center | Head | Head |

11 selected human joints, S is the scaling factor, and T is the transfer matrix which sets joint 4 (torso) as the origin of the system.

$$d_i = \frac{j_i}{S} + T, j_i \in J \tag{2}$$

The calculation method of scale factor S is the ratio of the distance between joint 4 and joint 2 in the current frame (from the torso to the neck) and a standard value of the distance between those two joints, named as h.

$$S = \frac{||j_4 - j_2||}{h} \tag{3}$$

Although there is redundant information between the direction vector and the position vector when they represent a posture, each has its own advantages. The direction vector can ignore the recognition problem brought about by the different length of body parts between different people while the position vector can express a joint's position more accurately. In this paper, we will combine them into linear fusion features to represent a posture. The posture feature vector will contain 10 direction vector (each direction vector has 3-dimension data) and 11 position feature vector (each position vector has 3-dimension data), having 63-dimension data in total.

$$F = [D_1, D_2, D_3, \ldots, D_{10}, d_1, d_2, d_3, \ldots, d_{11}] \tag{4}$$

## 2.2 Key Posture Selection

The reason to select the key postures is that the length of posture sequence between different actions is usually inconsistent. To unify the length of posture sequence, we need to choose the k most key postures in each action to represent it.

The focus of selecting the key postures is to ensure that the selected postures are critical. When a posture is greatly different from its neighboring postures in the same posture sequence, it is considered significant to represent this action. On the contrary, if there is not much difference between the selected posture and its neighboring postures, it is obviously not so important as to represent this action. The Euclidean distance between the posture feature vectors can be used to express the difference of the postures. The pseudocode of selecting the key postures is as follows, in which, P is a posture; the input of the algorithm is the posture sequence A; K stands for the number of the key postures; the output is the posture sequence A' of the key postures, which only contains K key postures.

```
If (size(A)>K)
  While(size(A)!=K)
    i = arg min_{p_i ∈ A} D(P_i);
    A =cut(A, i);
Else If (size(A)<K)
  While(size(A)!=K)
    i = arg max_{p_i, p_{i+1} ∈ A} D(P_i, P_{i+1});
    A =add(A, i);
A'=A;
```

The algorithm is mainly divided into the "cut" part and the "add" part. When the length of A is bigger than K, the cut part is used to cut off the unimportant postures and keep the important ones. This is the most common situation, so the cut part is used mostly. Sometimes the length of A is smaller than K, and then the add part is used to add some postures. When the number of input variables is 1, the function D in the algorithm calculates the mean value of Euclidean distance between the posture and its neighboring postures. If the posture has only one neighboring posture, the function only calculates the Euclidean distance between the two. When the number of input variables is 2, the function calculates the Euclidean distance of the two postures. The function Cut is used to remove the ith posture from the posture sequence A; The function Add is used to add a new posture behind the ith posture, and its value is the average of posture i and posture i + 1. In this way, the posture sequence of an action has become one with k postures through the key position selection algorithm, which can be used to represent this action. The action feature vector A contains K posture features and has K*63-dimension data in total.

$$A = [F_1, F_2, F_3, \ldots, F_K] \tag{5}$$

### 2.3 Classification

In this paper, Support Vector Machine (SVM) is selected as the classification algorithm. The implementation method of multiclass SVM basically has two kinds: one-against-many and one-against-one. In [18], the authors found that the "one-against-one" is one of the most suitable for practical use. It is implemented in LIBSVM [19] and it is adopted in this paper.

## 3 Experimental Results

In order to prove the feasibility and effectiveness of the proposed algorithm, it was tested by the two public data sets KARD [20] and Florence3D [21].

### 3.1 KARD

Gaglio et al. [17] collected KARD dataset. This dataset was composed of 18 activities that could be divided into 10 gestures (horizontal arm wave, high arm wave, two hand wave, high throw, draw x, draw tick, forward kick, side kick, bend, and hand clap), and 8 actions (catch cap, toss paper, take umbrella, walk, phone call, drink, sit down, and stand up). Each activity was repeated three times by ten different individuals (nine males and one female) with ages ranging from 20 to 30 years and height from 150 to 185 cm. They also proposed some evaluation experiments, including three different ways of doing experiments and two modalities of dataset splitting. The experiments were as follows:

(1) Experiment A: One-third of the data of each activity was used for training and the rest for testing.
(2) Experiment B: Two-thirds of the data of each activity was used for training and the rest for testing.
(3) Experiment C: Half of the data of each activity was used for training and the rest for testing.

The activities constituting the dataset were split in the following groups:

(1) Gestures and actions.
(2) Activity set 1, Activity set 2, and Activity set 3 are listed in Table 2. Activity set 1 is the simplest one since it is composed of quite different activities while the other two sets include much similar actions and gestures.

**Table 2.** KARD activities organized into three activity sets with different levels of difficulty

| Activity set 1 | Activity set 2 | Activity set 3 |
|---|---|---|
| Horizontal arm wave | High arm wave | Draw tick |
| Two hand wave | Side kick | Drink |
| Bend | Catch cap | Sit down |
| Phone call | Draw tick | Phone call |
| Stand up | Hand clap | Take umbrella |
| Forward kick | Forward kick | Toss paper |
| Draw x | Bend | High throw |
| Walk | Sit down | Horizontal arm wave |

In order to determine the value of K, the number of the key postures, we used LOOCV methods that Gaglio et al. suggested. We found that when the value of K is 28, the recognition rate reached its best (99.26%), while it reached 95% in experiments by Gaglio et al. In order to overcome the influence of randomness, each experiment was repeated 10 times.

As shown in Table 3, the results of the proposed algorithm are better than those in [17] in terms of both gestures and actions sets. The smallest gap of the results between [17] and our algorithm is 4.9% in the experiment B of action set, and the largest gap is 10.6% in experiment A of gesture set. As for the same data set, the largest gap of the proposed algorithm between different experiments is 1.5% in experiment A and experiment B of action set, while in [17] it is 6.5% in experiment A and experiment B of gesture set.

**Table 3.** Accuracy (%) for the test in terms of gestures and actions separately

| | Gestures | | Actions | |
|---|---|---|---|---|
| | [17] | Proposed | [17] | Proposed |
| Experiment A | 86.5 | 97.1 | 92.5 | 98.4 |
| Experiment B | 93.0 | 98.4 | 95.0 | 99.9 |
| Experiment C | 86.7 | 97.2 | 90.1 | 99.2 |

As shown in Table 4, the results of the proposed algorithm are better than those in [17] in three activity sets. The smallest gap of the results between [17] and our algorithm is 0.8% in the experiment B of activity set 1, and the largest gap is 16.9% in experiment C of activity set 3. In terms of the same data set, the largest gap of the proposed algorithm between different experiments is 1.5% in experiment A and experiment B of activity set 3, while in [17] it is 7.8% in experiment B and experiment C of activity set 3. The largest gap of the proposed algorithm between different sets in the same experiment is 1.4% in activity set 2 and activity set 3 of experiment A, while in [17] it is 11.3% in activity set 1 and activity set 3 of experiment C.

**Table 4.** Accuracy (%) for the test using three different activity sets

| | Activity Set 1 | | Activity Set 2 | | Activity Set 3 | |
|---|---|---|---|---|---|---|
| | [17] | Proposed | [17] | Proposed | [17] | Proposed |
| Experiment A | 95.1 | 99.5 | 89.9 | 99.8 | 84.2 | 97.4 |
| Experiment B | 99.1 | 99.9 | 94.9 | 100 | 89.5 | 98.9 |
| Experiment C | 93.0 | 99.8 | 90.1 | 99.9 | 81.7 | 98.6 |

Therefore, compared with [17], the proposed algorithm has a higher recognition rate and is less affected by difference of experiments, thus presenting better performance in similar activities.

### 3.2 Florence3D

Seidenari et al. [20] collected Florence3D dataset, which contained nine total daily activities (wave, drink from a bottle, answer phone, clap, tight lace, sit down, stand up, read watch, and bow). Ten people were asked to repeat several times for each activity and a total of 215 samples were collected. Compared with KARD, this dataset was more challenging because the sequences were much shorter and both the interclass similarity and the intraclass variability increased. The experimental method on this dataset is "New Person" test. According to this method, one person from ten was picked out in turn for testing, the rest for training. The results are shown in Table 5.

When the value of K is 10, the maximum recognition rate reaches 90.4%. So the proposed algorithm has a higher recognition ability compared with [20] and [21]. Different from [20] and [21], our research uses both the direction features, which overcome the influence from different human sizes, and the position features, which express more details. Therefore, our method can present a human posture more effectively and robustly.

**Table 5.** Accuracy (%) for the test using "New Person" in Florence3D

| Method | Accuracy (%) |
|---|---|
| [20] | 82.0 |
| [21] | 87.04 |
| Proposed | 90.4 |

## 4 Conclusion

This paper proposes a new framework for RGBD data-based human action recognition. Specifically, the fusion feature vector composed of both the direction vector and the position vector is used to represent a human posture, and then an algorithm is presented to extract from an action the key postures which compose action features as input for classification and recognition by a multiclass Support Vector Machine. We evaluate the effectiveness of our technique using two different datasets, KARD and Florence3D. The results show that the proposed algorithm has a greater recognition ability and performs better in similar activities compared with the methods in some other researches. Yet, there are still some limitations of this research. When serious occlusion happens between joints or when a human is upside down, our system does not perform very well.

**Acknowledgment.** This work is supported by the National Natural Science Foundation of China under Grant No. 61403302 and the Fundamental Research Funds for the Central Universities No. XJJ2016029.

## References

1. Baak, A., et al.: A data-driven approach for real-time full body pose reconstruction from a depth camera. In: Fossati, A., Gall, J., Grabner, H., Ren, X., Konolige, K. (eds.) Consumer Depth Cameras for Computer Vision, pp. 1092–1099. Springer, London (2013)
2. Hadfield, S., Bowden, R.: Kinecting the dots: particle based scene flow from depth sensors. In: IEEE Proceedings, vol. 58, no. 11, pp. 2290–2295 (2011)
3. Bagdanov, A.D., Del Bimbo, A., et al.: Real-time hand status recognition from RGB-D imagery. International Conference on Pattern Recognition IEEE, pp. 2456–2459 (2012)
4. Berretti, S., Bimbo, A.D., Pala, P.: Superfaces: a super-resolution model for 3D faces. In: Fusiello, A., Murino, V., Cucchiara, R. (eds.) ECCV 2012. LNCS, vol. 7583, pp. 73–82. Springer, Heidelberg (2012). doi:10.1007/978-3-642-33863-2_8
5. Gasparrini, S., et al.: Performance analysis of self-organising neural networks tracking algorithms for intake monitoring using kinect. In: IET International Conference on Technologies for Active and Assisted Living. IET (2015)
6. Shotton, J., et al.: Real-time human pose recognition in parts from single depth images. Postgrad. Med. J. **56**(1), 1297–1304 (2011)
7. Yamato, J., Ohya, J., Ishii, K.: Recognizing human action in time-sequential images using hidden Markov model. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 379–385 (1992)
8. Kellokumpu, V., Pietikäinen, M., Heikkilä, J.: Human activity recognition using sequences of postures. IAPR Conference on Machine Vision Applications, pp. 570–573 (2005)
9. Scholkopf, B., Smola, A.J.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge (2001)
10. Yang, X., Zhang, C., Tian, Y.L.: Recognizing actions using depth motion maps-based histograms of oriented gradients. In: ACM International Conference on Multimedia, pp. 1057–1060. ACM (2012)

11. Oreifej, O., Liu, Z.: HON4D: histogram of oriented 4D normals for activity recognition from depth sequences. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 716–723. IEEE (2013)
12. Gan, L., Chen, F.: Human action recognition using APJ3D and random forests. J. Softw. **8** (9), 2238–2245 (2013)
13. Wang, J., Liu, Z., Wu, Y., Yuan, Y.: Mining actionlet ensemble for action recognition with depth cameras. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1290–1297 (2012)
14. Althloothi, S., et al.: Human activity recognition using multi-features and multiple kernel learning. Pattern Recogn. **47**(5), 1800–1812 (2014)
15. Ohn-Bar, E., Trivedi, M. M.: Joint angles similarities and HOG2 for action recognition. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 465–470. IEEE Computer Society (2013)
16. Zhang, Z., et al.: A novel method for user-defined human posture recognition using Kinect. In: International Congress on Image and Signal Processing, pp. 736–740. IEEE (2014)
17. Gaglio, S., Re, G.L., Morana, M.: Human activity recognition process using 3-D posture data. IEEE Trans. Hum.-Mach. Syst. **45**(5), 1–12 (2014)
18. Hsu, C.W., Lin, C.J.: Errata to "a comparison of methods for multiclass support vector machines". IEEE Trans. Neural Netw. **13**(4), 415–425 (2002)
19. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. **2**(3), 389–396 (2011)
20. Seidenari, L., et al.: Recognizing actions from depth cameras as weakly aligned multi-part bag-of-poses. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 479–485. IEEE Computer Society (2013)
21. Devanne, M., et al.: 3-D human action recognition by shape analysis of motion trajectories on riemannian manifold. IEEE Trans. Syst. Man Cybern. **45**(7), 1023–1029 (2014)