# Evaluation of a skeleton-based method for human activity recognition on a large-scale RGB-D dataset

**Enea Cippitelli*, Ennio Gambi*, Susanna Spinsante*, Francisco Flórez-Revuelta+**

*Dipartimento di Ingegneria dell'Informazione, Università Politecnica delle Marche,
Ancona, Italy I-60131 E-Mail: {e.cippitelli, e.gambi, s.spinsante}@univpm.it
+Department of Computer Technology, University of Alicante
P.O. Box 99, E-03080 Alicante, Spain E-Mail: francisco.florez@ua.es

**Keywords:** kinect, human action recognition, bag of key poses, temporal pyramid, NTU RGB+D dataset

## Abstract

Low cost RGB-D sensors have been used extensively in the field of Human Action Recognition. The availability of skeleton joints simplifies the process of feature extraction from depth or RGB frames, and this feature fostered the development of activity recognition algorithms using skeletons as input data. This work evaluates the performance of a skeleton-based algorithm for Human Action Recognition on a large-scale dataset. The algorithm exploits the bag of key poses method, where a sequence of skeleton features is represented as a set of key poses. A temporal pyramid is adopted to model the temporal structure of the key poses, represented using histograms. Finally, a multi-class SVM performs the classification task, obtaining promising results on the large-scale NTU RGB+D dataset.

## 1 Introduction

Many vision-based Human Action Recognition (HAR) algorithms have been proposed in the last years, mainly because they can have many different applications, from surveillance to human-computer interaction, including also the support of ageing in place in Active and Assisted Living (AAL) environments [1]. In AAL, HAR has been identified as one of the most important components. In more detail, vision-based HAR techniques, if compared to ambient or mobile activity recognition, can provide very detailed information about the context [2], and can be used to unobtrusively extract people's movements. The main drawback of vision-based solutions is related to privacy, which is a concern that can be partially overcome with the adoption of RGB-D sensors. Unexpensive RGB-D sensors, such as Microsoft Kinect, are less susceptible to variations in light intensity than RGB cameras [3], and they allow to achieve an higher level of privacy by using only the human silhouette extracted from depth data, or only the skeleton to represent a person [4].

RGB-D sensors can be used to detect dangerous events in AAL, which are mainly falls. As proposed in literature, the sensor can be placed in different setups, from the frontal view [5] to the top view [3], often exploiting ad-hoc algorithms to recognize people and detect a fall. RGB-D sensors can be used also to monitor specific activities, an example can be the development of drink and food intake monitoring systems [6], where Kinect is placed on the ceiling and can track the intake movements of a person.

On the other hand, a general HAR algorithm can recognize a set of activities, and the adoption of an RGB-D sensor enables the exploitment of different features to perform this task [7]. Since depth data enable an easier extraction of the human silhouette, some action recognition algorithms proposed the use of 3D silhouettes. Li et al. [8] developed a method which represents postures considering a bag of points extracted from the contours of 3D silhouette. The temporal relationship among the postures is modeled using an action graph, where each node represents a salient posture. Spatio-temporal depth subvolume descriptors have been proposed also in [9], where the polynormals, a group of hypersurface normals containing geometry and local motion information, are extracted from depth sequences. The combination of polynormals provides the Super Normal Vector (SNV), which is the final representation of the depth map. Spatio Temporal Interest Points (STIP) are often used to extract features from RGB images, but they have been also applied to depth data. In [10], the use of depth STIPs is combined with a descriptor containing the spatio-temporally windowed pixels within a 3D cuboid centered at the interest point. A codebook is built by clustering the identified cuboids and an action can be represented as a sequence of elements from the codebook. Oreifej and Liu [11] proposed the HON4D descriptor, which is based on the orientations of normal surfaces in 4D. Being a holistic descriptor, it provides a representation for the entire sequence, not for the skeleton frame.

Skeleton joints are extracted from depth data, and can be seen as a compact representation of the human body. The joints estimation algorithm may be affected by noise, especially if the sensor does not face the person but, in complex environments,

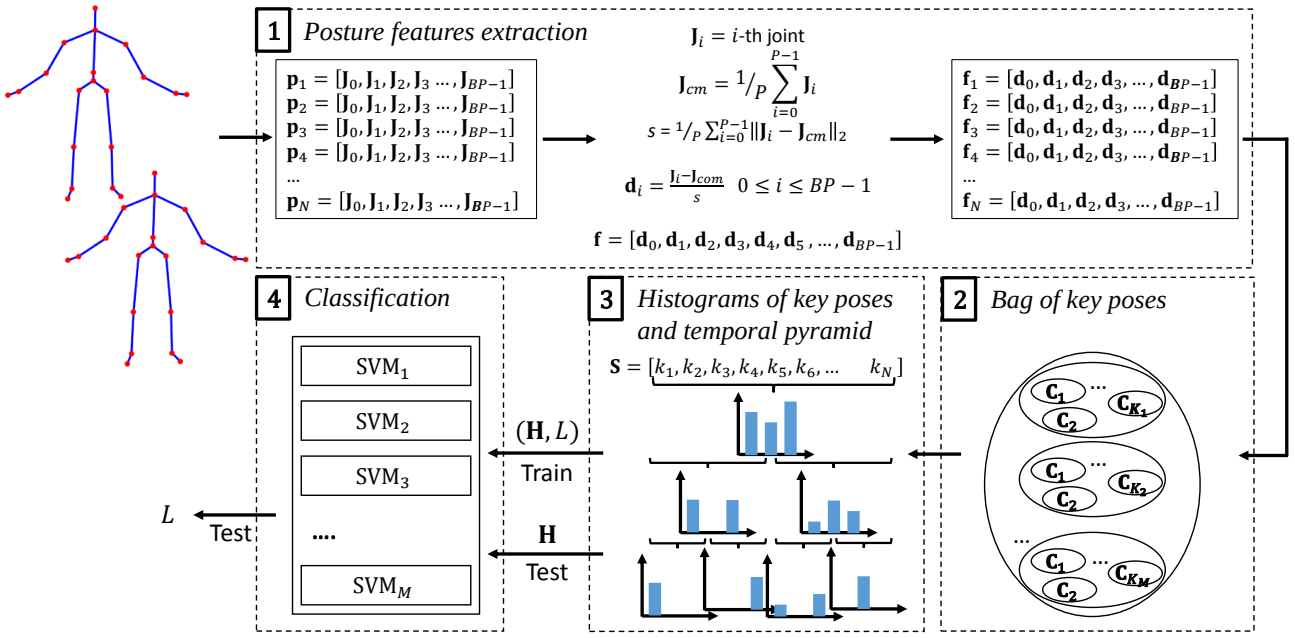


Figure 1. Global scheme of the activity recognition algorithm. The extraction of the posture features vector is the first step (block 1), followed by the generation of the codebook and the extraction of key poses (block 2). A temporal pyramid is adopted to keep the temporal structure of the action, and a set of histograms of key poses are obtained at each level of the pyramid (block 3). Finally, the classification is performed using a multi-class SVM (block 4).

some HAR algorithms based on skeleton data may achieve better results than depth-based ones. Considering the joint coordinates, different feature extraction methods have been proposed [7]. Vemulapalli et al. [12] proposed to represent rotations and translations of rigid body transformations as points in a Special Euclidean group $SE(3)$. A skeleton can be seen as a point in the Lie group $SE(3) \times SE(3) \times \cdots \times SE(3)$, and a human action, constituted by a sequence of skeletons, is considered as a curve in this Lie group. Evangelidis et al. [13] proposed a descriptor called skeletal quads, which encodes the relations of joint quadruples. Then, the action is represented as a Fisher vector which is the input to a multi-class Support Vector Machine (SVM). Considering the HOJ3D representation [14], the 3D space is partitioned into bins and each skeleton joint is associated to each bin using a Gaussian weight function. After a reprojection of HOJ3D histograms using Linear Discriminant Analysis (LDA), the clustering operation allows to obtain a fixed number of postures. A discrete Hidden Markov Model (HMM) models the temporal evolution of the postures. Hu et al. [15] proposed the jointly learning of features extracted by different sources of information: RGB, depth and skeleton data. Regarding skeleton joints, they compute relative positions between pairs of joints and extract human pose and its dynamics considering the temporal Fourier pyramid and its gradient version.

The proposed HAR algorithm considers skeleton joints and extracts pose-related features, handling also actions performed by multiple people. The most informative postures, i.e. key-poses, are learned through unsupervised clustering generating a bag of key poses model [16]. An action is modeled as histograms of key poses, with the adoption of a temporal pyramid to keep the distribution of the key-poses within the action. A multi-class SVM is the considered classification algorithm. The proposed method has been evaluated on the large-scale NTU RGB+D dataset [17], reaching results comparable to other approaches exploiting skeleton-based features.

The paper is organized as follows: Section 2 describes the details about the algorithm for activity recognition. The experimental results are presented and discussed in Section 3, while Section 4 provides conclusions.

## 2 HAR algorithm based on histograms of key poses

The 3D coordinates of the skeleton joints are the input data of the proposed action recognition algorithm and the features representing a specific posture are initially computed, using a modified approach with respect to the one proposed in [18]. The features belonging to the training sequences are then clusterized and a key pose, represented by a cluster center, is associated to each feature vector. An action is represented as a sequence of key poses, and the histograms of key poses are generated for each level of the temporal pyramid. The obtained histograms represent the feature vector considered for classification by a multi-class SVM. The entire process may be represented by 4 main steps, which are sketched in Figure 1 and detailed as follows:

1. *Posture features extraction*: in this step the 3D coordinates of the joints are considered and the features representing each posture are computed;

2. *Bag of key poses*: the codebook is generated by applying a clustering algorithm to the training data, and a key pose is associated to each posture in the sequence;

3. *Histograms of key poses and temporal pyramid*: given the number of levels of the temporal pyramid, a sequence of key poses is represented as a set of histograms obtained for each level;

4. *Classification*: the histograms of key poses are classified using a multi-class SVM with the "one-versus-all" method.

The algorithm can handle the presence of more skeletons in the scene, and the extraction of features representing the posture consists in the evaluation of the normalized position differences among each joint and the center-of-mass of the main skeleton. Considering that the $i$-th joint of a skeleton is represented by a three-dimensional vector $\mathbf{J}_i$, a vector $\mathbf{p}_n$ stores all the coordinates for the $n$-th frame of an activity constituted by $N$ frames. A frame contains $B$ bodies, each of which is represented by $P$ joints. Differently from [18], where the coordinate space was centered in the joint of torso, a center-of-mass $\mathbf{J}_{cm}$ is computed considering the average 3D position of the main skeleton, constituted by $P$ joints:

$$\mathbf{J}_{cm} = \frac{1}{P} \sum_{i=0}^{P-1} \mathbf{J}_i \tag{1}$$

The normalization factor $s$, previously represented by the distance between neck and torso joints [18], is now computed considering the average distance among all the joints of the main body and its center-of-mass, as follows:

$$s = \frac{1}{P} \sum_{i=0}^{P-1} \|\mathbf{J}_i - \mathbf{J}_{cm}\|_2 \tag{2}$$

The position difference $\mathbf{d}_i$ is represented by the displacement between the $i$-th joint and the center-of-mass, considering the scaling factor. All the $B$ bodies have to be considered in the computation of position differences, according to (3):

$$\mathbf{d}_i = \frac{\mathbf{J}_i - \mathbf{J}_{cm}}{s}, \qquad i = 0, 1, \ldots, BP-1 \tag{3}$$

Using the position displacement and the normalization factor, the features are invariant to the position of the skeletons within the coverage area of the sensor, and also to the build of the subjects. The posture feature vector $\mathbf{f}_n$, associated to the $n$-th skeleton frame, is finally constituted by all the $BP$ differences:

$$\mathbf{f}_n = [\mathbf{d}_0, \mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_{BP-1}] \tag{4}$$

If the sequence contains only one skeleton, part of the feature vector will contain zeros, but this part is kept anyway, to ensure the same dimensionality of the vector.

The second step concerns the generation of the codebook, which consists in the extraction of the the most informative feature vectors, which are the key poses. This process starts with the application of $k$-means clustering algorithm to the feature vectors, considering separately the vectors belonging to different actions of the dataset. With $M$ classes, that are the $M$ different actions of the dataset, the vector $[K_1, K_2, \ldots, K_M]$ specifies the number of key poses for each class. Following this approach, all the training instances of the first class are clustered in $K_1$ key poses, represented by the cluster centers $[\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_{K_1}]$. The codebook is obtained by merging all the key poses obtained for each class. A key pose is associated to each posture feature vector that constitutes an action, by considering the closest one in terms of euclidean distance. An action, originally represented by a sequence of features vectors $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_{n_1}]$, is encoded by a sequence of key poses $\mathbf{S} = [k_1, k_2, \ldots, k_{n_1}]$. In the testing phase, the codebook is exploited to associate unseen feature vectors to learned key poses.

The step number 3 considers the creation of the histograms of key poses for each level of a temporal pyramid. A sequence of key poses $\mathbf{S} = [k_1, k_2, \ldots, k_{n_1}]$ is split into $2^{l-1}$ segments, if $l$ is the actual level of the pyramid. A histogram is built for each segment of the pyramid considering the number of occurrences of each key pose within the segment, normalized by the sequence length. The temporal pyramid can effectively represent the distribution of the key poses within the sequence. Each segment is split in two parts, moving from the top to the bottom of the pyramid, allowing to have different descriptions of the same sequence, from the most general to the most detailed one. Furthermore, the computation of the histograms at the $l$-th level of the pyramid can be efficiently obtained considering the sum of the corresponding segments at the level $l+1$. The final representation of the sequence is constituted by the histograms at each level of the pyramid. The vector $\mathbf{H}$ in Figure 1, denotes the concatenation of the 7 histograms obtained considering a temporal pyramid with 3 levels.

The classification step aims to associate each set of histograms $\mathbf{H}$, which represents an action, to the corresponding class label, and it is based on a SVM. There are two main strategies to obtain a multi-class classifier from binary SVMs: "one-versus-all" and "one-versus-one". Considering an $M$-classes classification task, the "one-versus-all" method exploits $M$ binary SVMs, each of which trained to distinguish between one class and the rest. The winner class is the one with highest probability. The "one-versus-one" method, on the other hand, considers a number of $M(M-1)/2$ binary classifiers to account for all the possible pairs of classes. Each classifier is trained to separate two classes and the final outcome is obtained with a voting strategy: the output class is the one that gets more votes. This work exploits the "one-versus-all" strategy implemented in LIBLINEAR [19] library.

## 3 Experimental results

The algorithm performance is evaluated on the NTU RGB+D dataset [17], which is at the authors' best knowledge, the largest dataset for 3D action recognition currently available. It has been captured to overcome the limitations of the existing datasets, providing more data that are required to develop algorithms closer to real conditions. The dataset is constituted by

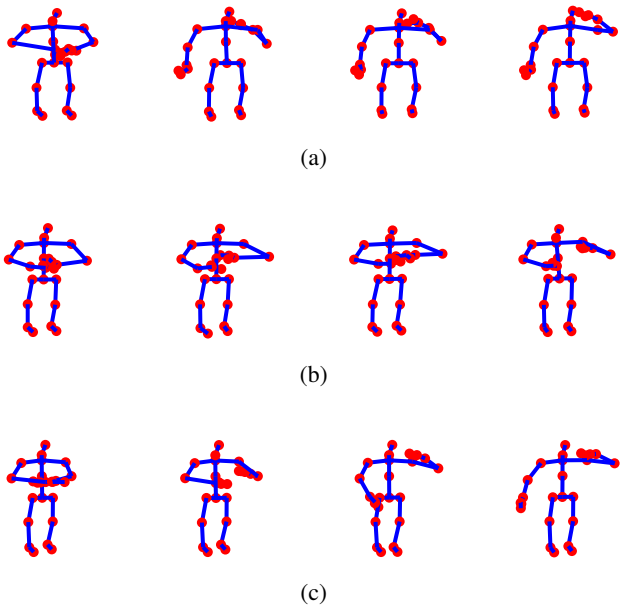

Figure 2. Sequences of frames constituting similar actions: (a) *drink water*, (b) *eat meal/snack* and (c) *brushing teeth*.

60 activities performed by 40 actors, aged between 10 and 35, and actions have been captured in 80 different views, resulting in a total number of 56880 sequences. Microsoft Kinect v2 has been used to collect the dataset, all the streams available from the sensor are captured and provided: depth and IR frames ($512 \times 424$), RGB images ($1920 \times 1080$) and 25 joints for each skeleton. The 60 activities can be grouped in 40 daily actions, 9 health-related actions and 11 interactions, where two skeletons are involved. In total, 17 different setups have been considered, each of which featuring different height of the cameras and different distances from the cameras to the subjects, within the interval $[2, 4.5]$ meters. Each action has been captured by three Kinect sensors at the same time, located at the same height with the horizontal angles $-45°$, $0°$, $+45°$. Each subject performed the action two times, once facing the left camera and once facing the right camera. Figure 2 show some frames belonging to the first 3 actions of the dataset: *drink water*, *eat meal/snack* and *brushing teeth*, performed by the same actor and captured by the frontal camera. The postures involved in those actions are very similar, this confirms that the dataset is quite challenging.

Shahroudy et al. [17] defined two evaluation methods, to enable a fair comparison of different activity recognition algorithms:

- cross-subject evaluation: the sequences performed by 20 actors are used as training and the others as testing data. The subjects that have to be used as training are: 1, 2, 4, 5, 8, 9, 13, 14, 15, 16, 17, 18, 19, 25, 27, 28, 31, 34, 35, 38. Considering that the authors recommend to discard 302 sequences featuring missing or wrong skeletons, the training and testing sets are respectively constituted by 40091 and 16487 samples (instead of 40320 and 16560).

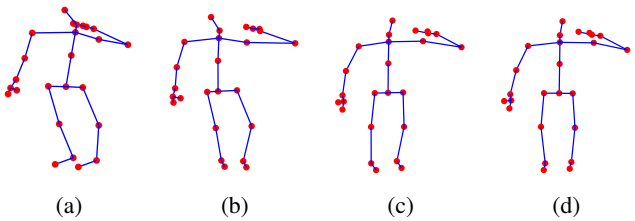

Figure 3. Original and rotated skeletons related to the same body during the capture of action 3 (*brushing teeth*). (a) and (b) show respectively the original and the rotated skeleton taken from camera 1, which has the $45°$ view. (c) and (d) show the same body obtained from camera 3, with the frontal view.

| | cross-subject | | | cross-view | | |
|---|---|---|---|---|---|---|
| $K$ | $L=2$ | $L=3$ | $L=4$ | $L=2$ | $L=3$ | $L=4$ |
| 4 | 41.8 | 45.2 | 43.8 | 44.3 | 49.1 | 50.1 |
| 8 | 44.5 | 48.1 | 45.0 | 49.3 | 53.0 | 52.5 |
| 16 | 46.4 | 48.7 | 45.1 | 52.3 | 56.3 | 54.9 |
| 32 | 48.1 | **48.9** | 46.8 | 55.1 | 57.2 | 56.2 |
| 64 | 47.7 | 48.3 | – | 55.3 | **57.7** | – |

Table 1. Recognition accuracy (%) obtained by the proposed method, with different number of clusters ($K$) and different levels of the temporal pyramid ($L$).

- cross-view evaluation: the sequences captured by different cameras are split between training and testing. The sequences from cameras 2 and 3 are used for training while data from camera 1 provide the testing set. After removing the 302 noisy sequences, the training set has 37646 samples (instead of 37920) and the testing set has 18932 samples (instead of 18960).

Before applying the activity recognition algorithm, the sequences have to be processed to apply rotation of the skeletons, to filter noisy data and to find the main actor. In order to compensate the effect of the different view points, the skeletons are rotated as suggested in [17], aligning the $X$ axis to the vector connecting the shoulders and the $Y$ axis to the vector from *spine base* to *spine* joints. Figure 3 shows the original and rotated skeletons related to the same body captured from two different view points during the *brushing teeth* action. The left part is related to camera 1, which observes the $45°$ view, while the right part is obtained from camera 3, which observes the front view. The pre-processing phase constituted by filtering of noisy data and identification of the main actor is implemented with a different strategy if the sequence belongs to training set or to testing set. In fact, during training phase, the activity is known, and it is possible to discern between action or interaction. If the activity is an action, only one skeleton has to be in the sequence. If there are more than one skeleton, the noisy ones have to be detected and removed. If all the skeletons of

the sequence are noisy, the frame is removed, while if none of them is noisy, the first one is kept. If the activity is an interaction, only two skeletons have to be in the sequence. If a sequence contains more than two skeletons, the filtering technique is implemented to find the noisy one. If none of them is a noisy skeleton, the first two skeletons are kept. The filtering technique suggested in [17] has been implemented, which consists in the removal of the skeletons whose joints show a spread over $X$ axis higher than 0.8 of the one over $Y$ axis. The standard deviation has been chosen to measure the spread. A sequence featuring two skeletons has to be processed to find the main one, which is characterized by the highest amount of body motion. This task is accomplished considering the sum of the displacements of each joint from one frame to the next one. During testing phase it is not possible to establish a priori the number of skeletons of a sequence. The only analysis implemented is the filtering technique to remove noisy skeletons and the assumption that the maximum number of skeletons is 2. So, even if there are 3 skeletons in the sequence and they are not noisy, the first two skeletons are kept, and the main one is the skeleton featuring the highest amount of body motion.

Tests have been executed considering $C = 1$ for the SVM, different levels of the temporal pyramid ($L = [2, 3, 4]$), and the same number of clusters for each action of the dataset, setting $K = K_1 = K_2 = \cdots = K_M$. Values within the interval $[4, 8, 16, 32, 64]$ have been investigated, and the obtained results for cross-subject and cross-view evaluations are shown in Table 1. Cross-subject evaluation is more challenging than the cross-view one, and the best results are represented by a large number of clusters. Considering the cross-subject test, the proposed algorithm obtained the best accuracy of $48.9\%$ with $K = 32$, which means a total number of key poses of 1920 for the 60 activities of the dataset. However, the accuracy is quite close to the best one also with a lower number of clusters: considering $K = 8$ the obtained performance is $48.1\%$. An higher accuracy has been obtained considering cross-view evaluation, where the best performance is given by the use of 64 key poses per class. In this configuration, the gap between a lower number of key poses and the maximum performance is higher: using $K = 4$ the accuracy is $49.1\%$, with a difference of $8.6\%$. The best results for cross-subject and cross-view evaluation schemes were both obtained with 3 levels for the temporal pyramid. The adoption of a simpler temporal pyramid with only 2 levels may lead to comparable results with a large number of clusters ($K = 32$ or $K = 64$), especially for the cross-subject test, while a larger gap is present comparing different levels with a small number of clusters ($K = 4$ or $K = 8$). The choice of 4 levels for the temporal pyramid increases the complexity of the algorithm but does not give better results in terms of accuracy, with the exception of the cross-view evaluation with $K = 4$, where the results for $L = 4$ are $1\%$ better than those for $L = 3$.

Table 2 shows the performance obtained by the proposed method, in comparison with previously published works evaluated on the cross-subject and cross-view tests. To the best of our knowledge, there are no other works using the NTU RGB+D dataset, and Table 2 has been obtained considering the tests run by Shahroudy et al. [17]. The proposed method does not reach results comparable to techniques based on deep neural networks but can be considered as an alternative method among the skeleton-based solutions.

| Method | cross-subject | cross-view |
|---|---|---|
| *Depth-based* | | |
| Oreifej and Liu [11] | 30.56 | 7.26 |
| Yang and Tian [9] | 31.82 | 13.61 |
| Ohn-Bar and Trivedi [20] | 32.24 | 22.27 |
| *Skeleton-based* | | |
| Evangelidis et al. [13] | 38.62 | 41.36 |
| **This method** | 48.9 | 57.7 |
| Vemulapalli et al. [12] | 50.08 | 52.76 |
| Hu et al. [15] | 60.23 | 65.22 |
| *Deep neural networks* | | |
| Du et al. [21] | 59.07 | 63.97 |
| Shahroudy et al. [17] | **62.93** | **70.27** |

Table 2. Comparison of different methods evaluated on NTU RGB+D dataset in terms of recognition accuracy (%). Results are ordered considering the accuracy of the cross-subject test.

## 4 Conclusion

In this work, a HAR algorithm based on skeleton joints has been evaluated on a new large-scale dataset, the NTU RGB+D one. The algorithm is based on bag of key poses model and exploits features which are invariant to build and position of the subjects. The temporal structure of the action is represented considering histograms of key poses at different levels of a temporal pyramid, and a multi-class SVM is adopted for classification. The proposed method achieves mid state-of-the-art results.

Future works will concern the evaluation of a combined cross-view-and-subject evaluation, which could verify the success of normalization process regarding the actor and the viewpoint at the same time.

## Acknowledgements

The authors would like to acknowledge the contribution of the COST Action IC1303 AAPELE (Architectures, Algorithms and Platforms for Enhanced Living Environments).

## References

[1] R. Poppe. A survey on vision-based human action recognition. *Image and Vision Computing*, 28(6):976 – 990, 2010.

[2] P. Rashidi and A. Mihailidis. A Survey on Ambient-Assisted Living Tools for Older Adults. *Biomedical and Health Informatics, IEEE Journal of*, 17(3):579–590, May 2013.

[3] S. Gasparrini, E. Cippitelli, S. Spinsante, and E. Gambi. A Depth-Based Fall Detection System Using a Kinect® Sensor. *Sensors*, 14(2):2756–2775, Feb. 2014.

[4] J. R. Padilla-López, A. A. Chaaraoui, F. Gu, and F. Flórez-Revuelta. Visual Privacy by Context: Proposal and Evaluation of a Level-Based Visualisation Scheme. *Sensors*, 15(6):12959–12982, 2015.

[5] G. Mastorakis and D. Makris. Fall detection system using kinect's infrared sensor. *Journal of Real-Time Image Processing*, 9(4):635–646, 2014.

[6] S. Gasparrini, E. Cippitelli, E. Gambi, S. Spinsante, and F. Flórez-Revuelta. Performance Analysis of Self-Organising Neural Networks Tracking Algorithms for Intake Monitoring Using Kinect. In *1st IET International Conference on Technologies for Active and Assisted Living (TechAAL)*, 2015.

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

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

[9] X. Yang and Y. Tian. Super Normal Vector for Activity Recognition Using Depth Sequences. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[10] L. Xia and J. K. Aggarwal. Spatio-temporal Depth Cuboid Similarity Feature for Activity Recognition Using Depth Camera. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2834–2841, June 2013.

[11] O. Oreifej and Z. Liu. HON4D: Histogram of Oriented 4D Normals for Activity Recognition from Depth Sequences. In *2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 716–723, June 2013.

[12] R. Vemulapalli, F. Arrate, and R. Chellappa. Human Action Recognition by Representing 3D Skeletons as Points in a Lie Group. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 588–595, June 2014.

[13] G. Evangelidis, G. Singh, R. Horaud, et al. Skeletal Quads: Human Action Recognition Using Joint Quadruples. In *Proceedings of the 22nd International Conference on Pattern Recognition (ICPR 2014)*, 2014.

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

[15] J. F. Hu, W. S. Zheng, J. Lai, and J. Zhang. Jointly learning heterogeneous features for rgb-d activity recognition. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5344–5352, June 2015.

[16] A. A. Chaaraoui, P. Climent-Pérez, and F. Flórez-Revuelta. Silhouette-based human action recognition using sequences of key poses. *Pattern Recognition Letters*, 34(15):1799–1807, 2013.

[17] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang. NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[18] E. Cippitelli, S. Gasparrini, E. Gambi, and S. Spinsante. A Human Activity Recognition System Using Skeleton Data from RGBD Sensors. *Computational Intelligence and Neuroscience*, vol. 2016, Article ID 4351435, 14 pages, 2016.

[19] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[20] E. Ohn-Bar and M. M. Trivedi. Joint Angles Similarities and HOG2 for Action Recognition. In *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 465–470, June 2013.

[21] Y. Du, W. Wang, and L. Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1110–1118, June 2015.