Human Action Recognition Based on Temporal Pyramid of Key Poses Using RGB-D Sensors

Enea Cippitelli[1], Ennio Gambi[1], Susanna Spinsante[1], and Francisco Florez-Revuelta[2]

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

Abstract. Human action recognition is a hot research topic in computer vision, mainly due to the high number of related applications, such as surveillance, human computer interaction, or assisted living. Low cost RGB-D sensors have been extensively used in this field. They can provide skeleton joints, which represent a compact and effective representation of the human posture. This work proposes an algorithm for human action recognition where the features are computed from skeleton joints. A sequence of skeleton features is represented as a set of key poses, from which histograms are extracted. The temporal structure of the sequence is kept using a temporal pyramid of key poses. Finally, a multiclass SVM performs the classification task. The algorithm optimization through evolutionary computation allows to reach results comparable to the state-of-the-art on the MSR Action3D dataset.

Keywords: kinect, human action recognition, bag of key poses, temporal pyramid, evolutionary computation

1 Introduction

Human Action Recognition (HAR) is an active research topic in computer vision, mainly because it may enable and facilitate different applications. Automatic action recognition algorithms can be, for example, applied in video-surveillance of public spaces, or in Active and Assisted Living (AAL) environments, to support ageing in place of older people [1, 2]. Another interesting application is represented by Human-Computer Interaction (HCI), where gesture recognition in particular can provide an efficient way to interface a system [3].

In this scenario, the availability of inexpensive depth sensors, such as Microsoft Kinect, has fostered the research exploiting 3D data, which presents some advantages with respect to RGB cameras, such as less susceptibility to variations in light intensity [4]. Furthermore, depth data allow the extraction of skeleton joints [5], and enable the exploitation of different features for action

recognition [6]. Many algorithms for action recognition exploiting 3D silhouettes have been proposed, since depth data make the process of silhouette extraction easier. Li et al. [7] developed a method that represents postures considering a bag of 3D points extracted from depth data. Only a small set of 3D points is considered, and a method has been developed to sample the representative 3D points by performing planar projections of the 3D depth map and extracting the points that are on the contours. Other interesting features are represented by local Spatio Temporal Interest Points (STIPs) applied to depth data [8]. Depth-based STIPs include a noise suppression scheme which can handle some characteristics of the depth images, such as the noise in the borders of an object, where the depth values show a big difference in the transition from foreground to background, or the noise given by errors in the depth estimation algorithm, which can result in some gaps in the depth map.

Despite the proposal of different depth-based descriptors, the skeleton joints extracted by depth data represent a compact and effective description of the human body, and many activity recognition algorithms rely only the joints as input. Considering joint coordinates, different feature extraction methods have been proposed. Some of them consider only spatial data, some others include also temporal information [6]. The HOJ3D representation [9] considers the partition of the 3D space into bins and the joints are associated to each bin using a Gaussian weight function. The histograms are clustered to obtain the salient postures and a discrete Hidden Markov Model (HMM) is employed to model the temporal evolution of the postures. In addition to k-means clustering, the use of sparse coding has been also proposed for the creation of the codebook. In particular, Luo et al. [10] proposed the DL-GSGC scheme, where the discriminative capacity of the dictionary is improved by adding group sparsity and geometry constraints to the sparse coding representation. A temporal pyramid is adopted to model the temporal information, and a linear Support Vector Machine (SVM) is chosen as the classification algorithm. Wang et al. [11] firstly considered relations among body joints in the spatial domain, by grouping joints into different body parts. Then, the temporal relations of the body parts are obtained, and actions are represented by histograms of the detected part-sets.

Feature selection methods or optimization strategies may be adopted to improve the performance of HAR algorithms. These methods may increase the recognition performance because they can select the relevant features for an efficient discrimination among the activities. Eweiwi et al. [12] proposed a HAR algorithm exploiting joints where the pose feature is a weighted sum of all joint features. The weights are estimated by Partial Least Squares (PLS). Wang et al. [13] proposed a data mining solution to discover discriminative actionlets, which are structures of base features built to be highly representative of one action and highly discriminative compared to other actions. Evolutionary computation has been successfully adopted in feature selection problems, and it has also been considered for the optimization of HAR algorithms [14]. Usually, two models are used to apply the evolutionary computation: the filter model and the wrapper model. The former determinates the features relevance consider-

ing their intrinsic properties, without including the learning method. The latter approach encloses the induction algorithm and, even if more computationally expensive, it is usually preferred because of better results [15]. Another model of evolutionary optimization is the coevolutionary algorithm, which considers several populations: individuals in a population are awarded fitness values based on their interactions with individuals from other populations. Interactions can be competitive, where individuals are rewarded at the expense of those with which they interact, or cooperative, where individual are rewarded if they work well with other individuals [16]. Cooperative coevolutionary algorithms have been also applied to address feature and parameter selection problems in HAR [17].

The HAR algorithm herein proposed considers skeleton joints and extracts features representing the person's posture. A bag of key poses model [18] is adopted, where the most informative postures are learned using the k-means clustering algorithm. Then, an action is modeled as histograms of key poses, and the temporal structure of the action is kept using a temporal pyramid. A multi-class SVM is finally exploited for classification. The algorithm parameters are optimized using evolutionary and cooperative coevolutionary algorithms proposed in [14] and [17], which detect the best configuration of joints, key poses, and training instances. The proposed algorithm reaches results comparable to the state-of-the-art on the well known MSR Action3D dataset [7].

The paper is organized as follows: Section 2 describes the proposed activity recognition algorithm, providing implementation details from the features computation procedure to the classification scheme. The optimization process by evolutionary computation is described in Section 3, and experimental results are presented and discussed in Section 4. Finally, Section 5 provides concluding remarks.

2 HAR algorithm based on temporal pyramid of key poses

The action recognition algorithm takes the 3D coordinates of the skeleton joints as input data and initially computes some position displacements between them, as the features representing a specific posture. All the feature vectors are then clustered to extract a set of key poses per action, which are then combined into a bag of key poses. Then, an action is represented as a sequence of key poses, from which histograms are computed. Histograms of key poses are then organized considering more levels of the temporal pyramid. The obtained histograms represent the input to a multi-class SVM, which performs the classification task. The entire process may be represented by 4 main steps, which are sketched in Fig. 1 and detailed in the following:

1. *Extraction of posture features*: in this step the 3D coordinates of the joints are considered and the features representing each posture are computed;
2. *Codebook generation and key poses substitution*: this phase consists of the codebook generation and the association of a key pose to each posture in the sequence;

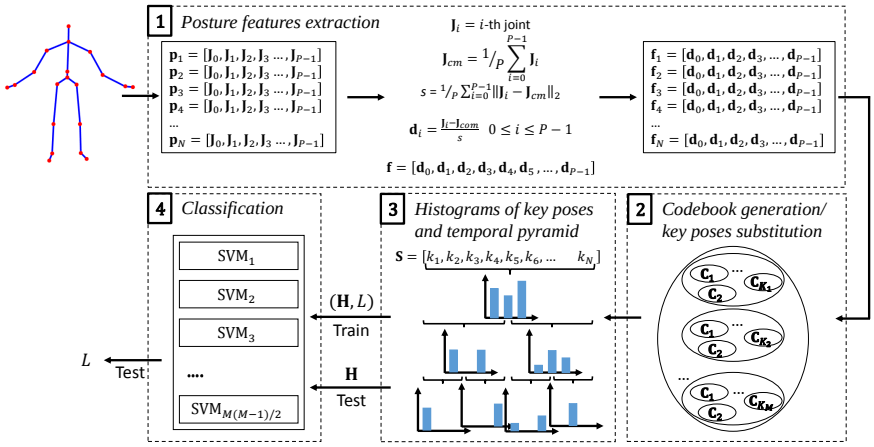


Fig. 1. Global scheme of the activity recognition algorithm. The first step consists in the extraction of the posture features vector, which are organized in a codebook to obtain the key poses. A sequence of key poses is then represented as a set of histograms obtained at each level of a temporal pyramid. Finally, the classification is performed using a multi-class SVM.

3. *Histograms of key poses and temporal pyramid*: a sequence of key poses is represented as a set of histograms obtained at different levels of a temporal pyramid;
4. *Classification*: the histograms of key poses are classified using a multi-class SVM, implemented following the "one-versus-one" method.

The extraction of features representing the posture consists of the calculation of the normalized position differences among the joints and their center-of-mass. Position differences are more robust features if compared to distances, with less ambiguity among different poses. 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 skeleton frame of an activity constituted by N frames. Each frame is represented by P joints, and the center-of-mass $\mathbf{J}_{cm}$ is represented by the average 3D position of all the P joints:

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

The normalization factor s is computed based on the average ℓ_2-norm between each joint and the center-of-mass, according to (2):

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

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, and it is implemented

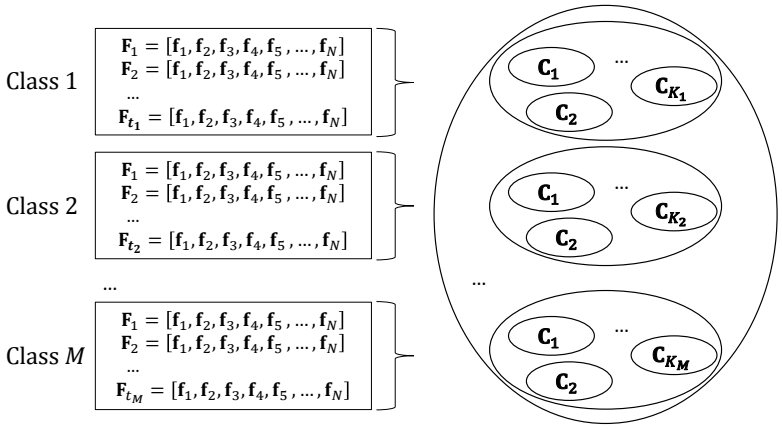


Fig. 2. Codebook generation and key poses extraction.

according to (3):

$$\mathbf{d}_i = \frac{\mathbf{J}_i - \mathbf{J}_{cm}}{s} \tag{3}$$

Using the difference between two positions makes the feature vector invariant to the position of the person within the 3D space, and the normalization by the scaling factor ensures the invariance to the build of the person. The feature vector $\mathbf{f}_n$, associated to the n-th skeleton frame, is finally made by all the differences for the P joints:

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

Due to errors in the skeleton estimation algorithm, the joints could be unavailable for some frames within the sequence. A skeleton integrity check is included in the feature extraction process and, if all the skeleton joints are unavailable for a specific frame, the posture feature vector related to the most recent skeleton frame is considered, and associated also to the actual frame.

The second step concerns the generation of the codebook, which contains the key poses, i.e. the most informative feature vectors. This process is implemented according to the k-means algorithm, by a separated clustering process for each action of the dataset. This choice is motivated by the fact that different actions may be better represented by a different number of key poses [14]. Considering M classes, that are the M different actions of the dataset, it is necessary to define a vector $[K_1, K_2, \ldots, K_M]$ containing the number of key poses for each class. The clustering process is sketched in Fig. 2, where, for example, all the training instances of the first class $[\mathbf{F}_1, \mathbf{F}_2, \ldots, \mathbf{F}_{t_1}]$ 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. Each feature vector in an action is finally substituted with the corresponding key pose, by considering the closest one in terms of Euclidean distance. At the end of this step, an action, previously represented by a sequence of feature vectors $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_{n_1}]$, is

encoded by a sequence of key poses $\mathbf{S} = [k_1, k_2, \ldots, k_{n_1}]$. Obviously, the codebook is generated during the training phase and exploited during testing, where unseen feature vectors are associated to learned key poses.

The third step regards the creation of the histograms of key poses obtained at each level of a temporal pyramid. The temporal pyramid is an effective representation to describe the temporal structure of a sequence representing an action. A sequence of key poses $\mathbf{S} = [k_1, k_2, \ldots, k_{n_1}]$ is split into 2^{l-1} segments, being l the level in the pyramid. For each segment, a histogram is obtained by counting the number of appearances of each key pose within the segment, and normalizing it to the segment length. The distribution of the key poses within the sequence is well represented by the temporal pyramid. Each segment is split into two parts, moving from the top to the bottom of the pyramid allows to have different descriptions of the same sequence, from the most general to the most detailed one. The final representation of the sequence is constituted by the histograms at each level of the pyramid. Considering a temporal pyramid of 3 levels, the whole sequence is represented by 7 histograms, denoted by the vector $\mathbf{H}$ in Fig. 1, containing the normalized number of occurrences for the 7 segments.

The last 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. SVMs have been originally defined as binary classifiers, and the most common approach to have a multi-class SVM is to combine many binary SVMs, with two options: "one-versus-all" and "one-versus-one". Considering an M-classes classification task, the former considers the definition of M binary SVMs, each of which is trained to distinguish between one class and the rest. The winner class is the one with highest probability. The "one-versus-one" method considers a number of $M(M-1)/2$ binary classifiers, each of which has to deal with two classes. The classification is done through a voting strategy, where all the classifiers select one class and the one with more votes is the output class. The "one-versus-one" method implemented in LIBSVM [19] is the one used in this work.

3 Optimization

The algorithm detailed in the previous section requires several parameters in order to be executed. These parameters can be heuristically chosen, but the use of an optimization algorithm may lead to better results. In HAR, evolutionary computation has been successfully used for feature selection and parameters optimization [14] [17]. The idea is to optimize three parameters of the HAR algorithm: the *features*, to select the optimal set of joints, the number of *clusters* to be used for each class in the bag of key poses model, and the set of training *instances*.

Considering the evolutionary optimization, the individual is constituted by three parts, each of them related to a different parameter. A detailed definition of the individual's structure can be found in [20], where the authors applied the evolutionary algorithm to have an evolving bag of key poses model. In this work, the same structure of the individual is exploited, where the *features* item is rep-

resented by a binary vector of length P, the *clusters* item is constituted by M integer values (one for each class), and the *instances* sub-individual is made up of I elements, each of them corresponding to a specific training sequence. Since the individual consists of three different parts, a 1-point crossover operator is applied to each part. A standard crossover is applied to *instances* and *clusters* vectors while a specific one, which is aware of the skeleton structure, is adopted for the *features* part. The mutation operator is also applied independently on the three parts of the individual with three probabilities mut_I (*instances* vector), mut_M (*clusters* vector) and mut_P (*features* vector). For the binary parts of the individual, each gene can change its value according to a mutation probability. Considering the *clusters* vector, the mutation is performed by considering a random value within an interval. The fitness value is represented by the accuracy of the HAR algorithm, and it is exploited to rank the individuals of the population.

In the cooperative coevolutionary algorithm, three different populations are defined: the *instances* population, the *clusters* population and the *features* one [17]. Each individual of the population has the same structure of the corresponding sub-individual considered in the evolutionary optimization, and the same choices about crossover and mutation operators can be adopted. In order to obtain a fitness value for a new individual of one population (i_1), it is necessary to consider also individuals from the two other populations (i_2 and i_3), and their selection is based on ranking. The obtained fitness value is updated for the individual i_1, but it is also updated for i_2 and i_3 if it improves their actual fitness value. Some techniques have been also adopted to give different priorities in the selection of individuals with the same fitness value. In *features* and *instances* populations, individuals with a lower number of selected values are preferred, while in the *clusters* population the individual with less accumulated sum is favored.

4 Experimental results

The performance of the algorithm has been evaluated on the MSR Action3D dataset [7], which is one of the most used datasets for action recognition. It is constituted by 20 activities performed by 10 actors, 2 or 3 times. In total, 567 sequences of depth (320×240) and skeleton frames are collected using a structured-light depth camera at 15 fps. Considering the skeleton frames, there are 557 sequences effectively available because 10 instances are featured by missing skeletons or they are affected by too many errors. The following activities are included in the dataset: *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, pickup and throw.* Due to its complexity, the dataset is usually evaluated considering three different subsets, namely AS1, AS2, and AS3 [7]. Padilla-López et al. [21] reviewed the papers based on the MSR Action3D dataset for action recognition and found that the most used evaluation scheme is the cross-subject test defined by Li et al. [7], which considers actors 1-

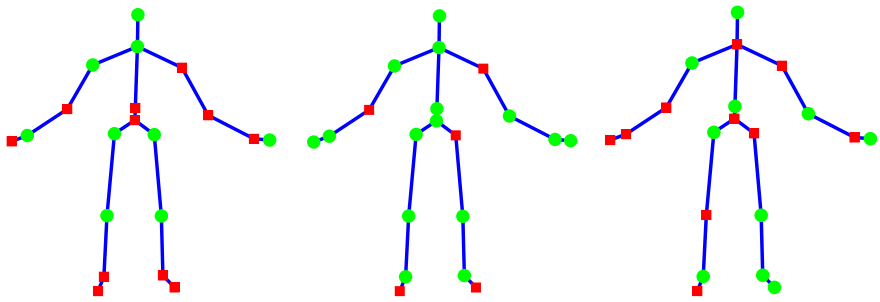
Table 1. Results obtained considering Random selection, Evolutionary and Coevolutionary optimizations.

	AS1	AS2	AS3
		Random selection	
Acc.	95.24	86.61	95.5
clust.	[17 17 15 25 8 22 12 22]	[4 8 10 22 18 19 16 5]	[71 66 48 56 66 61 76 52]
		Evolutionary optimization	
Acc.	95.24	90.18	100
clust.	[10 26 12 10 17 22 10 10]	[7 13 10 5 9 16 23 17]	[68 69 60 62 55 48 75 60]
feat.	[11100001011110001000]	[11100111110110011111]	[10100101110010100011]
		Coevolutionary optimization	
Acc.	95.24	91.96	98.2
clust.	[15 7 9 12 12 13 5 10]	[10 10 10 5 13 4 10 16]	[51 15 16 34 29 56 55 43]
feat.	[10101001100010001100]	[00001001101110011110]	[11111001011110100011]
inst.	178/219	202/228	176/222

3-5-7-9 for training, and actors 2-4-6-8-10 for testing. This evaluation procedure has also been applied in this work.

The selection of parameters for Radial Basis Function (RBF) kernel of SVM has been performed considering grid search and 5-fold cross-validation on training data, assuming the following intervals: $C = \left[2^{-5}, 2^{-3}, \dots 2^{15}\right]$ and $\gamma = \left[2^{-15}, 2^{-13}, \dots 2^{3}\right]$. The selection of parameters for the HAR algorithm has been performed using three different methods, all of them considering three levels of the temporal pyramid, with the following settings:

- *Random selection*: all the training instances and the features are considered, the clusters required by the bag of key poses model are selected randomly in the interval $[4, 26]$ for the subsets AS1 and AS2, while the interval $[44, 76]$ has been considered for AS3;
- *Evolutionary optimization*: all the training instances are considered, and the evolutionary algorithm is applied to select the features and the clusters, considering the same selection interval as the *Random* method. The population is constituted by 10 individuals, and the mutation probabilities have been randomly selected within the intervals $[0, 0.15]$ for mut_P and $[0, 0.25]$ for mut_C. The selection intervals for the clusters vector are the same as the *Random selection*, and the stop condition is reached after 100 generation without changing the best fitness value.
- *Coevolutionary optimization*: the optimization is applied to select instances, features and clusters, the mutation probability of instances vector mut_I is selected within the interval $[0, 0.025]$, and the clusters are randomly selected considering the interval $[4, 16]$ for AS1 and AS2, and $[4, 64]$ for AS3;



(a) (b) (c)

Fig. 3. Subsets of joints selected by the evolutionary algorithm for AS1 (a), AS2 (b) and AS3 (c). The selected joints are depicted as green circles, while the discarded ones are represented by red squares.

Table 1 shows the results obtained with the evolutionary and coevolutionary algorithms as optimization methods. Considering the optimization with the evolutionary algorithm, the optimized parameters are the number of clusters per class, and the set of skeleton joints that have to be selected. The performance obtained confirms that AS3 is the easiest subset to be recognized, and the proposed method can reach 100% score even if it requires a large number of key poses, which can be even 75 for the *golf swing* action. On the other hand, the set of selected features is rather limited, because only 10 joints out of 20 are required. AS2 is the most challenging subset, the best recognition accuracy is 90.18%, it requires a set of 15 joints and a reduced number of clusters, which is 23 at most. The algorithm requires only 9 joints and a restricted number of clusters also for the AS1 subset, where the recognition accuracy is 95.24%. Considering the joint representation in the *feature* vector, the selected subsets of joints by the evolutionary optimization is shown in Fig. 3. The coevolutionary optimization leads to the same average results. Considering AS1, the recognition accuracy is exactly the same, but only a number of 178 training instances are required out of the 219. Better results have been obtained considering AS2, the recognition accuracy of 91.96% is achieved with only a number of 10 joints and 202 training instances. Regarding AS3, the best accuracy obtained is 98.2%, and it is a suboptimal result that could be improved with a different stop condition, considering a greater number of iterations.

Table 2 shows the performance obtained by the proposed method, compared to the main HAR algorithms evaluated on the cross-subject test as well. The proposed method achieves results comparable to the state-of-the-art according to the accuracy averaged on AS1, AS2 and AS3 subsets. Shahroudy et al. [31], and Xu et al. [30] reach better average results but they exploit also depth data.

Table 2. Recognition accuracy (%) obtained by the proposed method, compared with other previously published works evaluated on the cross-subject test.

Method	AS1	AS2	AS3	avg
Li et al. [7]	72.9	71.9	79.2	74.67
Akkaladevi et al. [22]	84	62	80	75.3
Xia et al. [9]	87.98	85.48	63.46	78.97
Ghorbel et al. [23]	83.08	79.46	93.69	85.41
Evangelidis et al. [24]	88.39	86.61	94.59	89.86
Chen et al. [25]	96.2	83.2	92	90.47
Chaaraoui et al. [18]	92.38	86.61	96.4	91.8
Lo Presti et al. [26]	90.29	**95.15**	93.29	92.91
Tao and Vidal [27]	89.81	93.57	97.03	93.5
Du et al. [28]	93.3	94.64	95.5	94.49
Chen et al. [29]	98.1	92	94.6	94.9
This method	95.24	90.18	**100**	95.14
Xu et al. [30]	**99.1**	92.9	96.4	96.1
Shahroudy et al. [31]	–	–	–	**98.2**

5 Conclusion

In this work, a HAR algorithm based on skeleton joints has been proposed. A feature extraction scheme, which is invariant to build and position of the human subject has been exploited, and key poses are extracted from posture feature vectors. An effective representation of the action is obtained considering histograms of key poses at different levels of a temporal pyramid. The parameters optimization based on the evolutionary computation allows to reach results comparable to the state-of-the-art on the challenging MSR Action3D dataset. Future works include the use of a class-aware algorithm to estimate the key poses.

Acknowledgment

This work was supported by a STSM Grant from 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*, vol. 28, no. 6, pp. 976–990, 2010.

2. A. A. Chaaraoui, P. Climent-Pérez, and F. Flórez-Revuelta, "A review on vision techniques applied to human behaviour analysis for ambient-assisted living," *Expert Systems with Applications*, vol. 39, no. 12, pp. 10873–10888, 2012.
3. D. Weinland, R. Ronfard, and E. Boyer, "A survey of vision-based methods for action representation, segmentation and recognition," *Computer Vision and Image Understanding*, vol. 115, no. 2, pp. 224–241, 2011.
4. S. Gasparrini, E. Cippitelli, S. Spinsante, and E. Gambi, "A depth-based fall detection system using a kinect® sensor," *Sensors*, vol. 14, no. 2, pp. 2756–2775, Feb. 2014.
5. J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from a single depth image," in *CVPR*. IEEE, June 2011.
6. J. Aggarwal and L. Xia, "Human activity recognition from 3d data: A review," *Pattern Recognition Letters*, vol. 48, pp. 70–80, 2014.
7. 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*, June 2010, pp. 9–14.
8. 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*, June 2013, pp. 2834–2841.
9. L. Xia, C.-C. Chen, and 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*, 2012, pp. 20–27.
10. J. Luo, W. Wang, and H. Qi, "Group sparsity and geometry constrained dictionary learning for action recognition from depth maps," in *2013 IEEE International Conference on Computer Vision*, Dec 2013, pp. 1809–1816.
11. C. Wang, Y. Wang, and A. L. Yuille, "An approach to pose-based action recognition," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, June 2013, pp. 915–922.
12. A. Eweiwi, M. S. Cheema, C. Bauckhage, and J. Gall, *Computer Vision – ACCV 2014: 12th Asian Conference on Computer Vision, Singapore, Singapore, November 1-5, 2014, Revised Selected Papers, Part V*. Cham: Springer International Publishing, 2015, ch. Efficient Pose-Based Action Recognition, pp. 428–443.
13. J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, 2012, pp. 1290–1297.
14. A. A. Chaaraoui, J. R. Padilla-López, P. Climent-Pérez, and F. Flórez-Revuelta, "Evolutionary joint selection to improve human action recognition with RGB-D devices," *Expert Systems with Applications*, vol. 41, no. 3, pp. 786–794, 2014.
15. E. Cantú-Paz, *Genetic and Evolutionary Computation – GECCO 2004: Genetic and Evolutionary Computation Conference, Seattle, WA, USA, June 26-30, 2004. Proceedings, Part I*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, ch. Feature Subset Selection, Class Separability, and Genetic Algorithms, pp. 959–970.
16. R. P. Wiegand, "An analysis of cooperative coevolutionary algorithms," Ph.D. dissertation, Fairfax, VA, USA, 2004, aAI3108645.
17. A. A. Chaaraoui and F. Flórez-Revuelta, "Optimizing human action recognition based on a cooperative coevolutionary algorithm," *Engineering Applications of Artificial Intelligence*, vol. 31, pp. 116 – 125, 2014.

18. A. A. Chaaraoui, J. R. Padilla-López, and F. Flórez-Revuelta, "Fusion of skeletal and silhouette-based features for human action recognition with rgb-d devices," in *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*, Dec 2013, pp. 91–97.
19. C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.
20. A. A. Chaaraoui and F. Flórez-Revuelta, "Adaptive human action recognition with an evolving bag of key poses," *IEEE Transactions on Autonomous Mental Development*, vol. 6, no. 2, pp. 139–152, June 2014.
21. J. R. Padilla-López, A. A. Chaaraoui, and F. Flórez-Revuelta, "A discussion on the validation tests employed to compare human action recognition methods using the MSR Action3D dataset," *CoRR*, vol. abs/1407.7390, 2014.
22. S. C. Akkaladevi and C. Heindl, "Action recognition for human robot interaction in industrial applications," in *2015 IEEE International Conference on Computer Graphics, Vision and Information Security (CGVIS)*, Nov 2015, pp. 94–99.
23. E. Ghorbel, R. Boutteau, J. Boonaert, X. Savatier, and S. Lecoeuche, "3d real-time human action recognition using a spline interpolation approach," in *Image Processing Theory, Tools and Applications (IPTA), 2015 International Conference on*, Nov 2015, pp. 61–66.
24. G. Evangelidis, G. Singh, and R. Horaud, "Skeletal quads: Human action recognition using joint quadruples," in *Pattern Recognition (ICPR), 2014 22nd International Conference on*, Aug 2014, pp. 4513–4518.
25. C. Chen, K. Liu, and N. Kehtarnavaz, "Real-time human action recognition based on depth motion maps," *Journal of Real-Time Image Processing*, pp. 1–9, 2013.
26. L. L. Presti, M. L. Cascia, S. Sclaroff, and O. Camps, "Hankelet-based dynamical systems modeling for 3d action recognition," *Image and Vision Computing*, vol. 44, pp. 29–43, 2015.
27. L. Tao and R. Vidal, "Moving poselets: A discriminative and interpretable skeletal motion representation for action recognition," in *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*, Dec 2015, pp. 303–311.
28. 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)*, June 2015, pp. 1110–1118.
29. C. Chen, R. Jafari, and N. Kehtarnavaz, "Action recognition from depth sequences using depth motion maps-based local binary patterns," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, Waikoloa Beach, HI, Jan. 2015, pp. 1092–1099.
30. H. Xu, E. Chen, C. Liang, L. Qi, and L. Guan, "Spatio-temporal pyramid model based on depth maps for action recognition," in *Multimedia Signal Processing (MMSP), 2015 IEEE 17th International Workshop on*, Oct 2015, pp. 1–6.
31. A. Shahroudy, T. T. Ng, Q. Yang, and G. Wang, "Multimodal multipart learning for action recognition in depth videos," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PP, no. 99, pp. 1–1, 2015.