# Convolutional Neural Network-Based Action Recognition on Depth Maps

Jacek Trelinski[1] and Bogdan Kwolek[1]

AGH University of Science and Technology, 30 Mickiewicza, 30-059 Krakow, Poland
http://home.agh.edu.pl/~bkw/contact.html

**Abstract.** In this paper, we present an algorithm for action recognition that uses only depth maps. We propose a set of handcrafted features to describe person's shape in noisy depth maps. We extract features by a convolutional neural network (CNN), which has been trained on multi-channel input sequences consisting of two consecutive depth maps and depth map projected onto an orthogonal Cartesian plane. We show experimentally that combining features extracted by the CNN and proposed features leads to better classification performance. We demonstrate that an LSTM trained on such aggregated features achieves state-of-the-art classification performance on UTKinect dataset. We propose a global statistical descriptor of temporal features. We show experimentally that such a descriptor has high discriminative power on time-series of concatenated CNN features with handcrafted features.

## 1 INTRODUCTION

Action recognition is an active research topic with plenty of potential applications [1,2]. Research has focused on extracting conventional RGB image sequences and handcrafted features. Compared to traditional RGB image sequences the depth maps offer range information and are less sensitive to illumination changes. However, most current approaches to action recognition on depth maps are based on handcrafted features [3], which in many scenarios can provide insufficient discriminative power. Typically, human actions are recognized by extracting spatio-temporal features that are classified by multi-class discriminative classifiers and/or extracting time-series that are classified by Dynamic Time Warping (DTW) algorithms or algorithms relying on generative statistical models such as Hidden Markov Models (HMMs). However, in case of use of DTW the classification time of image/depth map sequences can be signifficant, whereas HMMs require considerable amount of training data.

Typical activity recognition algorithm involves three main steps: feature extraction, quantization/dimension reduction and classification. Approaches based on depth maps perform recognition using features extracted from depth maps and/or skeleton features, which are provided by Kinect motion sensors. Designing both effective and efficient features of depth sequence representations for action recognition is not an easy task due to several reasons [4]. The main reason is that in contrast to typical color features, the depth maps do not have

as much texture. Typically, they are too noisy both spatially and temporally to apply gradient operators both in space and time or to compute the optical flow, which is very useful motion descriptor and has proven to be useful in action recognition on RGB image sequences [1]. Last but not least, action recognition is typically performed on depth maps acquired by a single depth sensor. Thus, body parts are typically occluded, which in general leads to poor robustness of global features [4]. In order to cope with the challenges mentioned above, the researches developed several features that are semi-local, highly discriminative and robust to occlusion [5].

Due to noisy character of depth maps that prevent applying local differential operators, the number of depth maps-based sequential approaches, which achieved competitive results in comparison to depth-maps or depth-maps space-time volume approaches is quite limited [6]. Since the skeleton data is one of the most natural features for modeling action dynamics from depth maps, the most successful approaches use skeleton information [7]. In [8] a feature called Histogram of 3D Joint Locations (HOJ3D) that encodes spatial occupancy information with regard to the skeleton root was proposed. The HOJ3D features are computed on action depth sequences, projected using LDA and then clustered into $k$ posture visual words, which represent the prototypical poses of actions. The temporal evolutions of such visual words are modeled by discrete HMMs.

In this work, we present an algorithm for action recognition that uses only depth maps. We propose a set of handcrafted features to describe person's shape in noisy depth maps. We show experimentally that combining features extracted by a convolutional neural network (CNN) and proposed features leads to better classification performance. We show experimentally that LSTM trained on such aggregated features achieves state-of-the-art classification performance on UTKinect dataset. We propose a global statistical descriptor of temporal features. We show experimentally that such a descriptor has high discriminative power on time-series of concatenated CNN features with handcrafted features.

## 2 Datasets and Relevant Work

Introduction of low-cost 3D depth cameras such as MS Kinect have created many opportunities for human motion analysis and activity recognition. Li et al. [9] introduced a method for recognition of human actions from depth map sequences. It uses 3D contour points and does not require joint tracking. At the beginning, depth maps are projected onto three orthogonal Cartesian planes, and then a number of points along the contours of such projections are sampled for each frame. The motion dynamics is modeled by means of an action graph, whereas a Gaussian Mixture Model is used to robustly capture the statistical distribution of the points. The evaluation of the method has been performed on an introduced dataset, which later became known as the Microsoft Research (MSR) Action3D dataset. Experimental results have shown that high recognition accuracy can be achieved by sampling only about 1% of 3D points from the depth maps.

The MSR Action3D dataset [9] consists of twenty different actions, performed by ten different performers with up to three different repetitions. This makes in

total 567 depth map sequences and each one contains depth maps and skeleton joint positions. As explained by the authors, ten sequences are not valid since the skeletons were either missing or wrong. The different actions are high arm wave, horizontal arm wave, hand catch, hammer, high throw, forward punch, draw X, draw tick, draw circle, two hand wave, hand clap, side-boxing, bend, side kick, forward kick, jogging, tennis swing, golf swing, tennis serve, pick up & throw. These gaming actions cover different variations of the motion of torso, arms and legs. The mentioned above actions are performed without any interaction with objects. Each subject is facing the Kinect and is positioned in the center of the scene. Two main challenges in action recognition arise due to the high similarity between different groups of actions and changes of the execution speed of actions.

The dataset is divided into three subsets of eight actions each, which are called AS1, AS2 and AS3. The AS1 and AS2 subsets group actions with similar movement, while AS3 subset groups more complex actions together. For each subset, there are three different tests, i.e., Test One (T1), Test Two (T2), and Cross Subject Test (CST). In the test T1, 1/3 of the subset is utilized as training and the rest as testing, whereas in the test T2, 2/3 of subjects are utilized as training and the rest ones are used as testing. In the CST test, half of the subset is employed as training and the rest as testing.

UTKinect dataset [10] contains actions of ten different people performing one of 10 actions (walk, sit down, stand up, pick up, carry, throw, push, pull, wave hands and clap hands) in an office setting. Each subject performs each action twice. The dataset contains 200 data sequences with depth information, RGB data and skeleton joint locations, which were recorded at 15 fps. The actions included in the discussed dataset are similar to those from MSR Action3D, but they present some additional challenges: the actions were registered from different views. What is more, there are occlusions caused by human-object interactions or by lack of some body parts in the camera's field of view. Thus, the discussed dataset is more challenging than MSR Action3D dataset due to viewpoint variation and absence of body parts in the camera's field of view.

As noticed in a recently published survey [3], among datasets utilized in evaluation of action recognition algorithms, MSR dataset and UTKinect dataset are the most popular and widely used. For MSR Action3D dataset, most of the studies follow the evaluation setting of Li et al. [9], such they first divide the twenty actions into three subsets AS1, AS2, AS3, each having eight actions. For each subset, the tests T1, T2 and CST are typically performed. Most papers report classification accuracy better than 90% in the first two tests. In the third test, however, the recognition performance is usually far lower. This means that many of these methods do not have good generalization ability when different performer is performing the action, even in the same environmental conditions. For instance, the method of Li et al. achieves 74.7% classification accuracy in the CST test, whereas 91.6% and 94.2% accuracies were achieved in tests T1 and T2, respectively.

As mentioned in Introduction, methods based on locations of the joints achieve far better classification performance than methods relying on depth

maps or points clouds [7]. However, as noted in [11], skeleton-based methods are not applicable for applications, where skeleton data is not accessible. Since our method uses depth data only, below we discuss only depth-based methods.

In [12], depth images were projected onto three orthogonal planes and then accumulated to generate Depth Motion Maps (DMMs). Afterwards, the histograms of the oriented gradients (HOGs) computed from DMMs were utilized as feature descriptors. In [13] another method with no dependence on the skeletal joints information has been proposed. In the discussed method, random occupancy pattern (ROP) features were extracted from depth map sequences and a sparse coding was employed to encode these features. In [14], the depth map sequence is divided into a spatiotemporal grid. Afterwards, a simple feature called global occupancy pattern is extracted, where the number of the occupied pixels is stored for each grid cell. In [15] depth cuboid similarity features (DCSF) are built around the local spatio-temporal interest points (STIPs), which are extracted from depth map sequences. A method proposed in [16] does not require a skeleton tracker and calculates a histogram of oriented 4D surface normals (HON4D) in order to capture complex joint shape-motion cues at pixel-level. Unlike in [12], the temporal order of the events in the action sequences is encoded and not ignored. A recently proposed method [11] utilizes three projection views to capture motion cues and then employs LBPs for compact feature representation. In a more recent method [17], recognition of human action from depth maps is done using weighted hierarchical depth motion maps (WHDMM) and three-channel deep convolutional neural networks (3ConvNets).

## 3 Shape Features for Action Recognition

In our work we employ both handcrafted features and features extracted by convolutional neural networks (CNNs). In this Section we explain how we have extracted handcrafted and CNN features on the depth maps from datasets utilized in this work. The action performers are extracted from the background in the utilized datasets so no preprocessing step was needed to delineate the person.

### 3.1 Handcrafted Features

Given that pixels with non-zero values represent the performer on depth maps, only pixels with non-zero values were utilized in calculation of handcrafted features. The first feature is the ratio of the area occupied by the performer to the size of the whole depth map. The next nine features are calculated on the basis of the coordinates of pixels with non-zero depth values for axis $x, y$ and $z$, i.e. coordinates of pixels belonging to the performer. Based on such pixel coordinates we calculated the following features: (i) standard deviation of the non-zero pixel coordinates for axis $x, y$ and $z$, respectively, (ii) skewness of the non-zero pixel coordinates for axis $x, y$ and $z$, and (iii) correlation between the non-zero pixel coordinates for axes $xy, xz$ and $yz$. The handcrafted features were determined on depth maps scaled-down to sizes $60 \times 60$. For each depth map all features were normalized to zero mean and unit variance within the whole map.

### 3.2 Learning Convolutional Neural Network-based Features

**Convolutional Neural Network.** Convolutional neural networks are a category of neural networks that have proven to be very effective in areas such as image recognition and classification [18]. A CNN consist of one or more convolutional layers, very often with a subsampling step, followed by one or more fully connected layers as in typical multilayer neural networks [19]. They are neural architectures that integrate feature extraction and classification in a single framework. The main advantage of CNNs is that they are easier to train and have fewer parameters than fully connected networks with the same number of hidden units. Like classical neural networks they can be trained with a version of the back-propagation algorithm.

In the proposed algorithms the input depth maps have size $60 \times 60$ pixels. The convolutional layer C1 consists of sixteen $5 \times 5$ convolutional filters that are followed by a subsampling layer. The next convolutional layer C2 operates on sixteen feature maps of size $28 \times 28$. It consists of sixteen $5 \times 5$ convolutional filters that are followed by a subsampling layer. It outputs sixteen feature maps of size $12 \times 12$. The next fully connected layer FC consists of 100 neurons. At the learning stage, the output of the CNN is a softmax layer with number of neurons equal to the number of actions to be recognized. Such a network has been learned on depth maps from training parts of depth map sequences. After the training, the layer before the softmax has been used to extract shape features. The shape features were then stored in feature vectors. Having on regard that a typical action sequence consists of a number of depth maps, which are represented by multidimensional vectors, the actions are represented as multidimensional, i.e.multivariate time-series, where on every time stamp (for single depth map) we have more than just one variable. Such multidimensional time-series are classified by algorithms, which are described in Section 4.

**Learning Convolutional Neural Networks.** The neural networks have been trained on depth maps of size $60 \times 60$. Initially we trained a CNN with single channel input map on all depth maps from training parts of datasets. As it turned out, better results can be achieved by CNNs that are trained on multi-channel depth maps. In the discussed representation of the action data, we determined pairs of depth images in such a way that first element of the pair is the current depth map and the second element is the next depth map. In other words, a single par consists of two consecutive depth maps from a given depth map sequence. The images from the pair were then stored in two channels of a three-channel data representation. The third channel contains the projected depth map onto an orthogonal Cartesian plane. This means that we generated a side-view of the depth map, which has then been scaled to size $60 \times 60$. In such a data representation the CNN network operates on 3-channel depth maps, where two maps are taken from the pair, whereas the third component is the projected depth map onto the orthogonal Cartesian plane. The size of the feature vector extracted by the CNN is equal to 100.

## 4 Action Classification

In this Section we explain how the actions are classified using the proposed shape features. In Subsection 4.1 we present action classification with logistic regression. Afterwards, in Subsection 4.2 we outline dynamic time warping, which has been used to classify actions represented as time-series of features extracted by the CNN. In last Subsection we outline the LSTM network, which has been used to classify actions on the basis of vectors of CNN features as well as vectors consisting of concatenated CNN features and handcrafted features.

### 4.1 Action Classification Using Global Statistical Description of Temporal Features

The handcrafted features, which describe person's shape in a single frame were stored in multidimensional vectors to represent actions. Given such a multidimensional vector, i.e. multivariate time-series, a global statistical description of temporal features has been calculated. For each time-series we calculated the mean, standard deviation and skewness. This means that a single action is represented by a vector of size thirty. Alternatively, at the frame-level we concatenated the handcrafted features with CNN features. Having on regard that the size of feature vector extracted by the CNN is 100, the size of the vector with concatenated handcrafted and CNN features has size 110. Thus, the size of the vector representing the action sequence is equal to 330. The recognition of actions has been achieved using classical logistic regression classifier [20].

### 4.2 DTW-based Action Classification

In time-series classification problem one of the most effective methods is the 1-NN-DTW, which is a special k-nearest neighbor classifier with $k = 1$ and a dynamic time warping for distance measurement. DTW is a method that calculates an optimal match between two given sequences [21]. The sequences are warped non-linearly in the time dimension to find the best match between two samples such that when the same pattern exists in both sequences, the distance is smaller. Denote $D(i, j)$ as the DTW distance between sub-sequences $x[1 : j]$ and $y[1 : j]$. Then the DTW distance between $x$ and $y$ can be computed by the dynamic programming algorithm using the following iterative equation:

$$D(i, j) = \min\{D(i-1, j-1), D(i-1, j), D(i, j-1)\} + |x_i, y_j| \qquad (1)$$

The time complexity of calculation of DTW distance is $O(nm)$, where $n$ and $m$ are the length of $x$ and $y$, respectively.

### 4.3 LSTM-based Action Classification

In this Subsection we describe action recognition algorithm that is composed of LSTM recurrent layers, which are capable of automatically learning and modeling temporal dependencies. Recently, such architectures demonstrated state-of-the-art recognition performance in speech recognition [22].

**Long Short-Term Memory units.** Unlike traditional neural networks, recurrent neural networks (RNNs), take as their input not just the current input example, but also what they perceived one step back in time. RNNs allow cycles in the network graph such that the output from neuron $n$ in layer $l$ at time step $t$ is fed via weighted connections to each neuron in the layer $l$ (including the neuron $i$) at time step $t + 1$. One of the main issues in RNN training is the vanishing gradient. In order to cope with this undesirable effect a variation of recurrent net with so-called Long Short-Term Memory units, or LSTMs [23], were proposed by the Hochreiter & Schmidhuber as a solution to the vanishing gradient problem. LSTMs are recurrent neural networks that contain a memory to model temporal dependencies in time-series. An LSTM uses a memory cell with a gated input, gated output and gated feedback loop. In such a cell, information can be stored in, written to, or read from a cell, like data in a computers memory. An additional enhancement is the use of bidirectional layers. In a bidirectional RNN, there are two parallel RNNs, where one of them reads the input sequence forwards and the other reads the input sequence backwards.

It is worth noting that an LSTM network can be discriminative or generative. This means that LSTM can be used for classification tasks or to generate similar sequences like the training samples. In this paper, we utilize the discriminative ability of the LSTM for classification of series of action features.

**Learning LSTMs.** The actions are classified by a neural network with one hidden layer of LSTM cells. The input layer of the neural network operates on a given number of the features per time step. This means that descriptors extracted by the CNN (or CNN features concatenated with the handcrafted features of person's shape) are provided to the LSTM network one at a time. Depending on the variant of the algorithm, there are 100 (CNN operating on single channel depth maps), or 110 (handcrafted shape features concatenated with CNN features) input neurons (one for each element in the descriptor), 50 memory blocks (each with a memory cell and an input, forget and output gate), and 10 or 20 output neurons (one for each action class). The output layer contains neurons that are connected to LSTM outputs at each time step. In a single time step the input neurons are activated with descriptor values. Afterwards, the memory cells and the gates determine activation values based on the input values and on previous memory cell states. Then, the activations computed in such a way propagate to the output layer, and the process described above is repeated for the next descriptor from the action sequence. Finally, the softmax activation function is applied for each output neuron. Owing to the softmax the sum of all outputs is equal to one.

## 5 Results and Discussion

The proposed framework has been evaluated on two publicly available benchmark datasets: MSR Action3D dataset and UTKinect dataset. The datasets were

chosen due to their popularities in action recognition community. In the evaluations and all experiments, we used 557 sequences of MSR Action3D dataset. Half of the subjects were used as training data and the rest of the subjects as test data. It is worth noting that the classification setting employs half of the subjects as the training data and the rest of them as test data, which is different in comparison to evaluations based on AS1, AS2 and AS3 data splits and averaging the classification accuracies over such data splits. The classification performances obtained in the discussed setting are lower in comparison to classification performances achieved in AS1, AS2, AS3 setting due to larger variations across the same actions performed by different subjects. The cross-subject evaluation scheme that was utilized in [15,17] has been adopted in all experiments. It is worth noting that this scheme is different from the scheme employed in [8], where more subjects have been utilized for the training.

Table 1 shows recognition performance on challenging UTKinect dataset that has been achieved by logistic regression classifier using shape features and global statistical descriptors of temporal features. As we can observe, the concatenation of CNN and handcrafted features at frame-level leads to better classification performance in comparison to algorithm using only CNN features.

Table 1: Recognition performance on UTKinect achieved by logistic regression classifier using frame-features and global statistical descriptors of temporal features.

| | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|
| CNN | 0.8804 | 0.8999 | 0.8804 | 0.8723 |
| CNN + handcrafted | 0.9130 | 0.9172 | 0.9130 | 0.9094 |

Table 2 shows recognition performance that has been obtained by 1-NN-DTW and LSTM classifiers using time-series of shape features. The first row in discussed table presents results that have been achieved by DTW calculating Euclidean distance on CNN features. The second row in Tab. 2 shows the recognition performance that has been obtained using the CNN features and the LSTM classifier. As we can notice, the LSTM classifier operating on CNN features achieves considerably better results in comparison to 1-NN-DTW classifier using CNN features. The best results on UTKinect dataset were achieved by the LSTM operating on concatenated CNN features and handcrafted features at

Table 2: Recognition performance on UTKinect dataset using CNN features, achieved by 1-NN-DTW and LSTM classifiers.

| features | classifier | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|---|
| CNN | DTW | 0.8804 | 0.9127 | 0.8804 | 0.8824 |
| CNN | LSTM | 0.9457 | 0.9532 | 0.9457 | 0.9455 |
| CNN + handcrafted | LSTM | 0.9565 | 0.9584 | 0.9565 | 0.9551 |

frame-level. The classification accuracy of the proposed LSTM-based algorithm for action recognition on depth maps is better in comparison to classification accuracy achieved by the state-of-the-art algorithm [17], which is also based on a CNN.

Table 3 presents the recognition performance of the proposed method compared with the previous depth-based methods on the UTKinect dataset. As we can notice, the proposed method outperforms both methods based on handcrafted features [15,24,25] as well as recently proposed methods that are based on deep convolutional neural networks. Our method algorithm considerably from the WHDMM+3DConvNets method that employs weighted hierarchical depth motion maps (WHDMMs) and three 3D ConvNets. The WHDMMs are employed at several temporal scales to encode spatiotemporal motion patterns of actions into 2D spatial structures. In order to provide sufficient amount of training data, the 3D points are rotated and then used to synthesize new exemplars. In contrast, we recognize actions using LSTM or DTW, which operate on CNN features, concatenated with handcrafted features. The improved performance of our method may suggest that the proposed method has better viewpoint tolerance than other depth-based algorithms, including [17].

Table 3: Comparative recognition performance of the proposed method and previous depth-based methods on the UTKinect dataset.

| Method | Accuracy [%] |
|---|---|
| DSTP+DSF [15] | 78.78 |
| Random Forests [24] | 87.90 |
| SNV [25] | 88.89 |
| WHDMM+3DConvNets [17] | 90.91 |
| **Proposed Method** | **95.65** |

Table 4 presents results that were achieved on MSR Action3D dataset. As we can observe, the best results were achieved by logistic regression classifier trained on the global statistical descriptor of time-series, consisting of vectors of concatenated CNN and handcrafted features.

Table 4: Recognition performance on MSR Action3D dataset using CNN features, CNN features concatenated with handcrafted features, which has been achieved by 1-NN-DTW, LSTM and logistic regression classifiers, respectively.

| features | classifier | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|---|
| CNN | DTW | 0.8109 | 0.8292 | 0.8109 | 0.8082 |
| CNN | LSTM | 0.7091 | 0.7106 | 0.7091 | 0.6978 |
| CNN + handcrafted | LSTM | 0.7309 | 0.7234 | 0.7273 | 0.7082 |
| CNN | logistic regression | 0.8254 | 0.8361 | 0.8255 | 0.8167 |
| CNN + handcrafted | logistic regression | 0.8472 | 0.8598 | 0.8473 | 0.8440 |

Table 5 shows the recognition performance of the proposed method compared with the previous depth-based methods on the MSR-Action3D dataset. The recognition performance of the proposed framework has been determined using the same experimental cross-subject setting as that in [26], where subjects 1, 3, 5, 7, and 9 were utilized for training and subjects 2, 4, 6, 8, and 10 were utilized for testing. As we can notice, the proposed method achieves better classification accuracy in comparison to methods proposed in [9,26], and it has worse performance in comparison to recently proposed methods relying both on handcrafted features [12,27] and features extracted by deep learning methods [17]. One of the main reasons for this is insufficient amount of training data. It is worth noting that method [17] uses synthesized training samples on the basis of 3D points. As shown in Tab. 5, on more challenging UTKinect dataset the discussed method [17] achieved worse results in comparison to results obtained by our algorithm.

Table 5: Comparative recognition accuracy of the proposed method and previous depth-based methods on the MSR-Action3D dataset.

| Method | Accuracy [%] |
|---|---|
| Bag of 3D Points [28] | 74.70 |
| Actionlet Ensemble [26] | 82.22 |
| Our Method | 84.72 |
| Depth Motion Maps [12] | 88.73 |
| Range Sample [27] | 95.62 |
| WHDMM+3DConvNets [17] | 100 |

The results presented above were achieved using CNNs trained on pairs of consecutive depth maps as well as depth map projections. Without the depth map projections the recognition accuracy of the algorithm is almost two percent smaller in comparison to algorithms not using depth map projections. The recognition accuracy of the algorithm using CNNs trained on single depth maps instead of pairs of consecutive depth maps is smaller more than seven percent.

The proposed method has been implemented in Python using Theano and Lasagne deep learning frameworks. The Lasagne library is built on top of Theano. The values of the initial weights in CNNs and LSTM networks were drawn randomly from uniform distributions. The cross-entropy loss function has been used in the minimization. The CNN networks were trained using SGD with momentum. The LSTM has been trained using backpropagation through time (BPTT) [29]. Much computations were performed on a PC computer equipped with an NVIDIA GPU card. The source code of the proposed algorithms is freely available[1].

[1] `https://github.com/tjacek/DeepActionLearning`

## 6 Conclusions

In this work a method for action recognition on depth map sequences using concatenated CNN features with handcrafted ones has been proposed. Due to considerable amount of noise in depth maps that prevent applying local differential operators, the number of depth maps-based sequential approaches is quite limited. We demonstrated experimentally that a sequential algorithm, in which an LSTM or a DTW operates on time-series of CNN features can achieve superior results in comparison to results achieved by state-of-the-art algorithms, including recently proposed deep learning algorithms. The method has been evaluated on two widely employed benchmark datasets and compared with state-of-the-art methods. We demonstrated experimentally that on challenging UTKinect dataset the proposed method achieves superior results in comparison to results achieved by recent methods. In comparison to recently proposed WHDMM+ 3DConvNets method [17] it achieves about 5% improvement in the recognition accuracy in the cross-subject evaluation scheme. In our experiments, data of subjects with even numbers was used for learning of the models, whereas data of subjects with odd numbers were utilized for testing the classifiers.

**Acknowledgment.** This work was supported by Polish National Science Center (NCN) under a research grant 2014/15/B/ST6/02808.

## References

1. Aggarwal, J., Ryoo, M.: Human activity analysis: A review. ACM Comput. Surv. **43**(3) (2011) 16:1–16:43
2. Malawski, F., Kwolek, B.: Real-time action detection and analysis in fencing footwork. In: 40th Int. Conf. on Telecomm. and Signal Proc. (TSP). (2017) 520–523
3. Liang, B., Zheng, L.: A survey on human action recognition using depth sensors. In: Int. Conf. on Digital Image Computing: Techniques and Appl. (2015) 1–8
4. Aggarwal, J., Xia, L.: Human activity recognition from 3D data: A review. Pattern Recognition Letters **48** (2014) 70 – 80
5. Chen, L., Wei, H., Ferryman, J.: A survey of human motion analysis using depth imagery. Pattern Recogn. Lett. **34**(15) (2013) 1995–2006
6. Ye, M., Zhang, Q., Wang, L., Zhu, J., Yang, R., Gall, J.: A survey on human motion analysis from depth data. In: Dagstuhl 2012 Seminar on Time-of-Flight Imaging and GCPR 2013 Workshop on Imaging New Modalities. LNCS, vol. 8200, Springer (2013) 149–187
7. Lo Presti, L., La Cascia, M.: 3D skeleton-based human action classification. Pattern Recogn. **53**(C) (2016) 130–147
8. Xia, L., Chen, C.C., Aggarwal, J.: View invariant human action recognition using histograms of 3D joints. In: CVPR Workshops. (2012) 20–27
9. Li, W., Zhang, Z., Liu, Z.: Action recognition based on a bag of 3D points. In: IEEE Int. Conf. on Computer Vision and Pattern Rec. - Workshops. (2010) 9–14
10. Xia, L., Chen, C., Aggarwal, J.: Human detection using depth information by Kinect. In: CVPR 2011 Workshops. (2011) 15–22

11. Chen, C., Jafari, R., Kehtarnavaz, N.: Action recognition from depth sequences using depth motion maps-based local binary patterns. In: 2015 IEEE Winter Conf. on Applications of Computer Vision. (2015) 1092–1099
12. Yang, X., Zhang, C., Tian, Y.L.: Recognizing actions using depth motion maps-based histograms of oriented gradients. In: Proc. of the 20th ACM Int. Conf. on Multimedia, ACM (2012) 1057–1060
13. Wang, J., Liu, Z., Chorowski, J., Chen, Z., Wu, Y.: Robust 3D action recognition with random occupancy patterns. In: Proc. of the 12th European Conf. on Computer Vision. LNCS, vol. 7573, Springer (2012) II:872–885
14. Vieira, A., Nascimento, E., Oliveira, G., Liu, Z., Campos, M.: STOP: Space-time occupancy patterns for 3D action recognition from depth map sequences. In: CIARP. Volume 7441 of LNCS., Springer (2012) 252–259
15. Xia, L., Aggarwal, J.: Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera. In: IEEE Int. Conf. on Computer Vision and Pattern Recognition. (2013) 2834–2841
16. Oreifej, O., Liu, Z.: HON4D: Histogram of oriented 4D normals for activity recognition from depth sequences. In: IEEE Int. Conf. on Computer Vision and Pattern Recognition. (2013) 716–723
17. Wang, P., Li, W., Gao, Z., Zhang, J., Tang, C., Ogunbona, P.: Action recognition from depth maps using deep convolutional neural networks. IEEE Trans. on Human-Machine Systems **46**(4) (2016) 498–509
18. Schmidhuber, J.: Deep learning in neural networks: An overview. Neural Networks **61** (2015) 85 – 117
19. LeCun, Y., Haffner, P., Bottou, L., Bengio, Y.: Object recognition with gradient-based learning. In: Shape, Contour and Grouping in Computer Vision. LNCS, vol. 1681, Springer (1999) 319–345
20. Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag (2006)
21. Paliwal, K., Agarwal, A., Sinha, S.: A modification over Sakoe and Chiba's dynamic time warping algorithm for isolated word recognition. Signal Processing **4**(4) (1982) 329 – 333
22. Sainath, T., Vinyals, O., Senior, A., Sak, H.: Convolutional, long short-term memory, fully connected deep neural networks. In: IEEE Int. Conf. on Acoustics, Speech and Signal Processing. (2015) 4580–4584
23. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8) (1997) 1735–1780
24. Zhu, Y., Chen, W., Guo, G.: Fusing multiple features for depth-based action recognition. ACM Trans. Intell. Syst. Technol. **6**(2) (2015) 18:1–18:20
25. Yang, X., Tian, Y.L.: Super normal vector for activity recognition using depth sequences. In: IEEE Int. Conf. on Computer Vision and Pattern Recognition. (2014) 804–811
26. Wu, Y.: Mining actionlet ensemble for action recognition with depth cameras. In: IEEE Int. Conf. on Computer Vision and Pattern Recognition. (2012) 1290–1297
27. Lu, C., Jia, J., Tang, C.: Range-sample depth feature for action recognition. In: IEEE Int. Conf. on Computer Vision and Pattern Recognition. (2014) 772–779
28. Ji, X., Liu, H.: Advances in view-invariant human motion analysis: A review. IEEE Trans. on Systems, Man, and Cybernetics, Part C **40**(1) (Jan 2010) 13–24
29. Werbos, P.: Backpropagation through time: what it does and how to do it. Proceedings of the IEEE **78**(10) (1990) 1550–1560