Real-time Multi-scale Action Detection From 3D Skeleton Data

Amr Sharaf, Marwan Torki, Mohamed E. Hussein*
Department of Computer and Systems Engineering
Alexandria University, Egypt
{amr.sharaf, mtorki, mehussein}@alexu.edu.eg

Motaz El-Saban
Microsoft Advanced Technology Lab Cairo
motazel@microsoft.com

Abstract

In this paper we introduce a real-time system for action detection. The system uses a small set of robust features extracted from 3D skeleton data. Features are effectively described based on the probability distribution of skeleton data. The descriptor computes a pyramid of sample covariance matrices and mean vectors to encode the relationship between the features. For handling the intra-class variations of actions, such as action temporal scale variations, the descriptor is computed using different window scales for each action. Discriminative elements of the descriptor are mined using feature selection. The system achieves accurate detection results on difficult unsegmented sequences. Experiments on MSRC-12 and G3D datasets show that the proposed system outperforms the state-of-the-art in detection accuracy with very low latency. To the best of our knowledge, we are the first to propose using multi-scale description in action detection from 3D skeleton data.

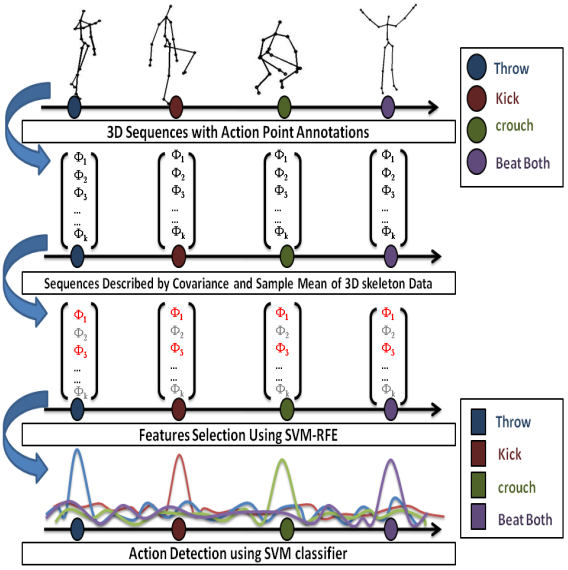

Figure 1. Overview of the action detection approach proposed in this paper.

1. Introduction

Robust action detection remains a very challenging computer vision task. Action detection entails the localization problem which is much more challenging than action recognition on temporally segmented sequences. The ability to detect human actions in real-time is fundamental to several applications such as surveillance, gaming, and sign language detection. These applications demand accurate and robust detection of actions at low latencies.

Action detection requires highly distinctive features to identify specific actions among many alternatives. The features must be invariant to common action variations such as temporal scale, translation, and illumination conditions. With the availability of body movement sensing technology, such as Microsoft kinect, it is now possible to perform pose estimation and capture the 3D skeleton data in real-time [21]. Compared to RGB images, 3D skeleton data is insensitive to changes in lighting conditions. In addition, features extracted from 3D joint positions of human skeleton are distinctive and can be used effectively for action detection.

In this paper, we present a novel method to detect human actions in 3D skeleton sequences (Figure 1). Our objective is not only to recognize the human action, but also to localize the recognized action in the video sequence. Specifically, we extract features from the 3D skeleton data and utilize multivariate statistical methods for encoding the relationship between the extracted features. A temporal pyramid of sample covariance matrices and sample mean vectors for the extracted features is used as a discriminative descriptor for each action.

We use a Support Vector Machine (SVM) classifier [8] with a linear kernel to perform action detection. Due to the

*Mohamed E. Hussein is currently an Assistant Professor at Egypt-Japan Univeristy of Science and Technology, on leave from his position at Alexandria University, where most of his contributions to this work were made.

high dimensionality of our descriptor, we use the Recursive Feature Elimination algorithm (SVM-RFE) [13] for selecting the most discriminative set of features. Our experiments show that only a small subset of the features are sufficient to outperform the state-of-the-art. Feature selection ensures that the action detection task can be performed in real-time.

Previous approaches to action detection lacked invariance to temporal action scales. For handling the variations in temporal scales, we propose a multi-scale action detector (Figure 2). Instead of using a sliding window of fixed scale, our detector takes a sequence of frames at different scales (i.e. different window sizes) and returns a probability confidence score for each action detected. The detector performs multi-scale detection and handles the necessary non-maximum suppression for merging nearby detections.

We emphasize our main contributions over prior work:

1. **Action Detection:** We address the problem of action detection, i.e. localizing and recognizing the class of an action in a video stream. This is different from the problem of action recognition, in which a wizard localizes the action and the problem reduces to recognizing its class. In the detection problem, the location of the action is marked by its action point, which is the point in time at which the application should react to the performance of the action.

2. **Real-time operation:** Typically, sliding window search is deployed over multiple scales, which mandates real-time performance of the detection algorithm. Our algorithm achieves 1140 fps. This is sufficient to apply an exhaustive sliding window search with up to 40 different scales in real-time.

3. **Multi-scale Detection:** Our detector accommodates for variations in action temporal scale by computing the descriptor for different window scales suitable for each action.

2. Related Work

In this section, we review recent related work for skeleton-based action detection and recognition as well as feature selection. The reader is referred to [1] and [20] for a more comprehensive analysis.

Skeleton-based Action Detection and Recognition We distinguish between the research on the two different problems: action detection and action recognition. Most of the literature focused on the action recognition problem. In action recognition, the video sequences are pre-segmented and the start and end of each action is pre-defined. On the other hand, the action detection problem requires the much more difficult task of localizing the detected action in an unsegmented stream of video frames.

In the context of action recognition, skeleton-based approaches have become widely used as Shotton *et al*. [21] proposed a real-time method to extract 3D positions of body joints from depth maps. Several descriptors in the literature proved how the position, motion, and orientation of joints could be excellent descriptors for human actions. Among these descriptors is the work of Ofli *et al*. [19] who computed a Sequence of Most Informative Joints (SMIJ) based on measures like the mean and variance of joint angles and the maximum angular velocity of body joints. Another work is done by Gowayyed *et al*. [11] where a descriptor named Histogram of Oriented Displacements (HOD) was introduced. Each displacement in the trajectory votes with its length in a histogram of orientation angles. Wang *et al*. [24] used relative skeleton position and local patterns to model the human-object interaction; an actionlet ensemble was used to model each action. More recently, [26] introduced a novel descriptor that is based on 3D kinematics in the joints motion. A mere drawback of using such descriptors is the high dimensionality in the descriptor to thousands of features.

Aside from the recent work in building efficient descriptors, there are efforts in using these descriptors in a learning framework to improve the accuracy. An example is the work by [17] where a sparse coding approach was applied to learn dictionaries for the action classes. Also the Ensemble work by Wang *et al*. [24] used ensemble learning.

Although the approaches mentioned above proved to be effective in regard to the action recognition problem, they haven't been extended to tackle the detection problem yet.

On the other hand, recent research focused on the action detection problem. Some approaches merely focused on detecting the start and the end points for the action without indicating the exact point in time where the action was performed. More advanced approaches perform accurate and real-time detection for actions. This is particularly useful in applications such as interactive gaming and sign language detection.

As an example for detecting the start and end points for an action, [26] proposed a non-parametric Moving Pose (MP) framework for action recognition. A moving pose descriptor that considers both pose information as well as differential quantities of the human body joints is computed. The authors extended the approach to solve the action detection problem on unsegmented video sequences. Another example is the work presented by [27], where they proposed a feature extraction method that uses a dynamic matching approach to construct a feature vector for each frame. The constructed feature vectors are used for detecting the start and end points for actions. However, both [26, 27] require providing a label for every frame during training and testing.

Fothergill *et al*. [10] was the first to introduce the no-

tion of an action point: a single time instance at which the presence of the action is clear. Action points allow a very accurate and fine-grained detection for the actions. In [10], a fixed sliding window approach of 35 frames was used for performing the action detection task. However, this approach fails to accommodate for actions of different temporal scales. Recently the action point notation was used in the work of [2] where they provided an annotation to the G3D dataset [3].

Thus, the approach we propose for action detection uses action points and provides a solution to the multi-scale problem. Additionally we perform feature selection to reduce the size for extracted features.

Feature Selection The existing algorithms can be grouped into three categories: filter methods, wrapper methods and embedded methods. Filter methods [12] select important features by measuring the correlation between the features and the classifier output. Wrapper methods [13, 6] rely on a learning algorithm to decide a subset of important features. Embedded methods [4, 5] embed the feature selection process into the classifier learning process. The SVM-RFE [13] algorithm is a wrapper method for performing feature selection. Due to its successful use in selecting informative genes for cancer classification, SVM-RFE gained great popularity [13, 22, 9]. Compared with other feature selection methods, SVM-RFE is an efficient and scalable wrapper method for performing feature selection.

3. Action Description

In this section, we explain the descriptor used to represent a sequence of frames containing an action. In the following section, we will explain how action detection is performed using a classifier trained on the descriptor outlined here. The steps of descriptor construction are illustrated in Figure 1.

3.1. Feature Extraction from 3D Skeleton Data

The body skeleton returned from a depth sensor consists of 3D coordinates of body joints (20 joints returned from the Kinect sensor). These coordinates are represented in the camera coordinate system. Therefore, they vary with the rotation and translation of the body with respect to the camera. To make our action descriptor relatively invariant to body rotation and translation, we use joint angles and angular velocities, which are derived from 3D joint coordinates. Particularly, we use the same 35 angles used in [18]. Twenty three of these angles are made by 3 different joints of the skeleton. The 12 remaining angles are made by 2 joints from the skeleton, and the camera itself, where the camera here can be seen as a virtual joint placed at the origin of the camera coordinate system.

To summarize, the 3D skeleton returned for the i^{th} frame is represented by a feature vector $x_i = [x_{i1}, x_{i2}, ..., x_{i,2K}]^T$, where K is the number of joint angles, which is 35 in our case. The first K elements of x_i are joint angles, and the last K elements are the corresponding angular velocities.

3.2. Descriptor Construction

Given N feature vectors, $x_1, x_2, ..., x_N$, representing a sequence of N frames, a descriptor for the action (if any) performed in the sequence has to be constructed. The descriptor should be representative for the action, and should have a fixed length, regardless of the number of frames, N.

The proposed descriptor is built from the parameters of the joint probability distribution of the feature vectors. Particularly, the feature vectors of the frame sequence are assumed to be independently drawn samples from some unknown probability distribution $\mathbf{p}(x)$. The sample mean vector, $\hat{\mu}$, and covariance matrix, $\hat{\Sigma}$, of $\mathbf{p}$ are computed from this random sample. In addition to the representativeness of these parameters to the performed action, the size of both parameters depends only on the number of features, not the number of frames. Hence, these parameters make a viable choice for action description.

The joint probability distribution of skeleton features in a sequence does not capture the temporal information, which is sometimes crucial in action recognition. To solve this issue, we adopt a temporal pyramid construction, which is commonly used in the literature. Particularly, a sequence of N frames is divided into possibly-overlapping sub-sequences of lengths $N/2$, $N/4$, ... Generally, the l^{th} level of the pyramid contains sub-sequences of length $N/2^{l-1}$ frames, and consecutive sub-sequences in the level are separated by a constant step. In our experiments, we compare two modes of the descriptor construction: the no-overlapping mode, and overlapping mode. For the no-overlapping mode, the l^{th} level of the pyramid contains 2^l non-overlapping sub-sequences. In the overlapping mode, the l^{th} level contains $2^{l+1}-1$ sub-sequences, in which case, the step separating two consecutive sub-sequences equals half of the sub-sequence's length.

After the mean vectors and covariance matrices are estimated for the entire sequence and all sub-sequences in the temporal pyramid, a single vector is assembled by stacking elements from all vectors and matrices into one long feature vector. Due to the symmetry of covariance matrices, only elements from the upper or the lower triangle are included.

The descriptor, as presented so far, contains elements that differ significantly in their typical ranges of values. For example, the range of an angle's mean may differ significantly from the corresponding angular velocity's mean. Also, the ranges of covariance and mean elements may be significantly different. Before training a classifier on the extracted descriptors, they have to be properly normalized so that the classifier can make effective use of all the elements. In our case, the descriptor is normalized such that each el-

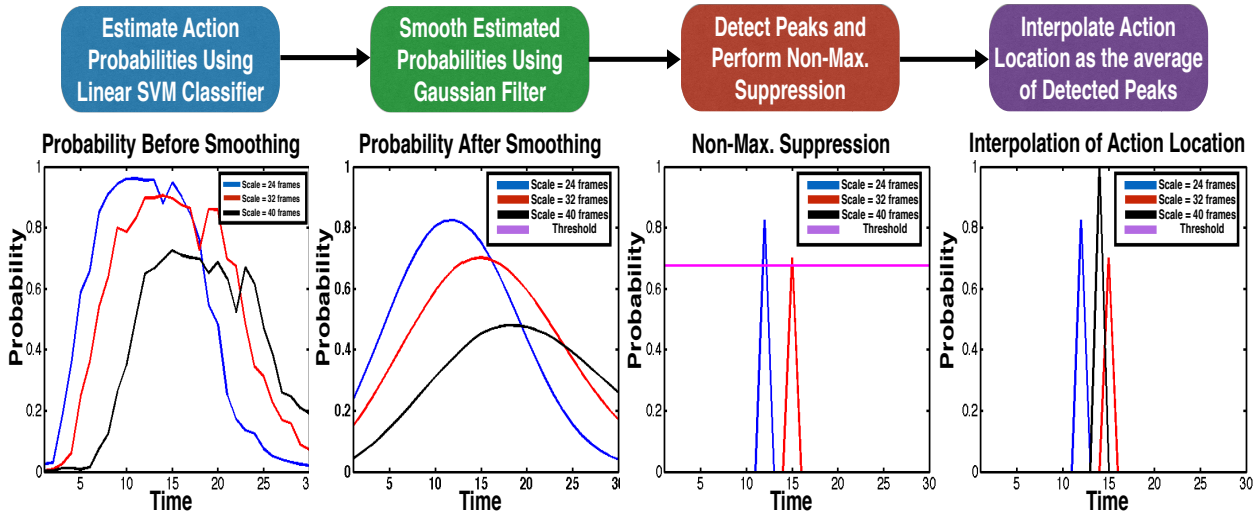

Figure 2. Overview of the multi-scale action detection approach proposed in this paper. The three curves in the figure are prediction probabilities for a single action class at three different scales.

ement of the final descriptor has a zero sample mean and unit sample standard deviation over the training samples. The mapping used to achieve this effect is then applied to the testing samples before evaluation.

3.3. Feature Selection with SVM-RFE

Feature selection is performed using SVM-RFE [13]. Selecting a subset of feature leads to a lower computation cost and enables the system to perform exhaustive sliding window search without compromising the real-time operation of the system.

4. Multi-scale Action Detection

Most of the previous approaches to action detection lacked handling of different action scales. In [10], a fixed sliding window of 35 frames is used for action detection. Also, in [26], a fixed sliding window size (W) was learned using cross-validation and used for action detection on unsegmented sequences. Even though good results are reported using this fixed window approach, better performance could be achieved by using an action detector that accommodates for different action scales. Different action classes typically have different scales. Even for a single action class, different subjects could perform it at different scales due to inter-subject differences in style and speed.

To address the aforementioned problem and for handling the variations in action scales, we propose a multi-scale action detector (Figure 2). Instead of using a sliding window of a fixed scale, our detector processes a sequence of frames at different scales (i.e. different window sizes), which depend on the action class.

Hussein *et al.* [14] provided ground truth annotations for the start and end frames of each action in the MSRC-12 dataset. Using these annotations, we analyzed the distribution of action scales. The scale is measured as the number of frames from the start of the action until the action point, which is the frame at which the action should be detected (details in Section 5.3). From this analysis, we select three scales for each action class: small (5^{th} percentile), medium (50^{th} percentile), and large (90^{th} percentile). When creating descriptors for the training data, the scale of each training sample is approximated to the closest of the three selected scales for the sample's action class while using the action point as the ending frame of the sample.

At testing time, at each frame t of the test sequence, for each action class and scale, a descriptor is constructed for the range of frames from $t-\sigma_{ci}+1$ to t, where σ_{ci} is the i^{th} selected scale for action class c. Descriptors are passed to the trained multi-class classifier, which returns a probability confidence score for each action class at each scale.

Because the returned probability scores are noisy, the raw probability scores are smoothed by being convolved with a Gaussian filter. To maintain the on-line operation of our approach, probability scores beyond the allowed observation latency (Δ) are not included in the convolution. Peaks of the smoothed probability scores that exceed a given threshold are detected. Then, non-maximum suppression is performed within the detection tolerance window (Δ). This is done separately for each action class and each scale. Finally, if multiple detections from multiple scales of the same action class are found within the detection tolerance window (Δ), their locations are averaged to find the final detected action location.

	Fothergill *et al*. [10]	Single Scale (2OL)	1 Level	2 Level - Overlap	3 Levels - No Overlap
Video - Text	0.679 ± 0.035	0.671 ± 0.129	0.677 ± 0.096	0.704 ± 0.146	0.713 ± 0.105
Image - Text	0.563 ± 0.045	0.652 ± 0.119	0.594 ± 0.106	0.670 ± 0.105	0.656 ± 0.122
Text	0.479 ± 0.104	0.508 ± 0.121	0.513 ± 0.075	0.516 ± 0.092	0.521 ± 0.072
Video	0.627 ± 0.052	0.686 ± 0.087	0.592 ± 0.075	0.637 ± 0.055	0.635 ± 0.075
Image	0.549 ± 0.102	0.583 ± 0.086	0.542 ± 0.092	0.579 ± 0.098	0.596 ± 0.103
Overall	0.579	0.620	0.584	0.621	0.624

Table 1. Results for MSRC-12 dataset. We report F-Score at $\triangle = 333ms$. We show the average and standard deviations over ten leave-persons-out runs. SVM-RFE algorithm is used to select 200 features only for the 1 level and 2 levels. For the 3 levels we use the 200 selected from the 2 levels with overlap settings and replicate them to have the descriptor size of 3 levels without overlap = 200×7.

5. Experimental Results

5.1. Evaluation Datasets

We evaluate our approach on two challenging datasets: MSRC-12 [10], and G3D [3].

The Microsoft Research Cambridge-12 dataset provides sequences of skeleton data, represented as 20 joint locations. The body poses were captured using the Kinect device at a sampling rate of 30 fps. The dataset contains 30 subjects performing 12 different gestures. The gestures are categorized into two categories: iconic and metaphoric gestures. The dataset was collected using five different types of instruction modalities: images, text, video, images + text, and video + text.

The G3D dataset contains a range of gaming actions captured with Microsoft Kinect. This dataset contains 10 subjects performing 20 gaming actions. These actions are grouped into seven categories: fighting, golf, tennis, bowling, FPS, driving, and miscellaneous actions.

5.2. Experimental Setup

The experiments were performed on both the MSRC-12 and the G3D datasets. We followed the same experiment setup as in [10], we used a "leave-persons out" protocol. For the MSRC-12 dataset, for each instruction modality, we remove a set of people from the full dataset to obtain the minimum test set that contains performances of all gestures. The remaining larger set is used for training. This is repeated ten times and the average test performance over the ten runs represents a good estimator of the generalization performance of the trained system. For the G3D dataset, all actors perform all the actions. We remove one subject from the full dataset to construct the test set and the larger remaining set of videos is used for training. This process is repeated 10 times with different subjects to obtain the general performance. We used a linear SVM classifier trained using the LIBSVM software [7].

5.3. Evaluation Metrics

Our proposed descriptor was evaluated using performance metrics designed for the task of real-time action detection. We followed the notion of "action points" as defined by [18]: "a single time instance at which the presence of the action is clear and that can be uniquely identified for all instances of the action". Action points enable latency-aware training and evaluation of real-time human action detection systems. Action point annotations are available for both the MSRC-12 as well as the G3D dataset.

The performance of the system is measured in terms of precision, recall, and latency. Precision indicates how often the gesture is actually present when the system claims it is. Recall measures how many true gestures are recognized by the system. The system latency shows how large the delay between the true action point and the system prediction is.

We follow the experimental setup used in [10] and [2]: for a specified latency tolerance ($\triangle ms$) we measure the precision and recall for each action $a \in A$. Where A is the set that contains all the different actions in the dataset. We combine both measures to calculate a single F-score [18] defined as: F-score$(a, \triangle) = 2\frac{prec_a(\triangle).rec_a(\triangle)}{prec_a(\triangle)+rec_a(\triangle)}$. Because the system detects multiple actions, we used the mean F-score over all action: F-score$(A, \triangle) = \frac{1}{|A|}\sum_{a \in A}$F-score$(a, \triangle)$

5.4. Real-time Action Detection Experiment

Human action recognition for pre-segmented video sequences is very useful for understanding the baseline performance for the recognition system. However, our main goal is to perform online action detection in real-time with low latency. Instead of determining the start and the end for each action sequence, we focus on the problem of detecting the action points within the video sequence.

An action point is detected by computing the probability for each class on each frame in the video sequence as presented in Figure 1 and comparing each probability to a threshold T. We used probability estimates from linear SVM classifier. To minimize computational latency, we used the SVM-RFE feature selection algorithm to find a subset of features that achieve the best possible performance. Multi-scale action detection is performed as described in section 4.

Action Type	Our Approach	Bloom *et al*. [2]
Fighting	0.937	0.919

Table 2. Results for G3D [3] dataset, We report F-Score at $\triangle = 333ms$. We show the average over ten leave-one-out runs.

5.4.1 Action Detection on MSRC-12 Dataset

The MSRC-12 Kinect Gesture dataset is designed to make the consideration of latency in human action detection possible. As described earlier, a specific amount of tolerated latency ($\triangle ms$) is selected, and the experiment measures whether the system can correctly recognize the gesture within a window of $\triangle ms$ before and after the gesture's action point. Table 1 shows the results for our proposed descriptor using different configurations. The results show that our descriptor outperforms the state-of-the-art-results [10] for all the five modalities used for collecting the dataset. The overall relative improvements reached up to 7.7% when compared to [10]. For "image and text" instruction modality relative improvements reached up to 16.5% and 8.8% for "text" modality.

To illustrate the importance of multi-scale action detection, we've also performed the detection experiment using a single scale of 35 frames (the same scale used by Fothergill *et al*. [10]). We used a 2 level descriptor with overlapping configuration. As show in table 1, using multi-scale detection outperforms the single scale approach. However, the improvement is not significant in the MSRC-12 dataset. This is justified since the scale statistics on this dataset revealed that most of the sequences had an average length of 35 frames.

5.4.2 Action Detection on G3D Dataset

Table 2 shows the results for the action detection experiment on the G3D dataset. Results show that we are able to outperform the state-of-the-art for action detection on the G3D dataset [3]. Results in [2] were reported for the "Fighting" action sequences only, however, we list our results for all seven action sequences in the supplementary materials. Results are reported using 200 features selected using SVM-RFE of 10220 features by using 2 Levels with overlap.

5.4.3 Action Recognition on Action3D Dataset

The MSR-Action3D dataset [15] consists of 557 segmented video sequences. 20 different action classes are performed by 10 subjects. The dataset focuses on action recognition on pre-segmented video sequences and is not suitable for the problem of real-time action detection on unsegmented videos. However, we're reporting the recognition results on the MSR-Action3D dataset to illustrate the discriminative power of our proposed descriptor used in the detection problem. We used a 3 level descriptor with overlapping configuration.

Method	Accuracy
Eigenjoints [25]	82.3%
Random Occupy Pattern [23]	86.2%
Actionlets Ensemble [24]	88.2%
Covariance Descriptor [14]	90.5%
Group Sparsity and Geometry Constrained Dictionary Learning [16]	96.7%
Our Approach	91.1%

Table 3. Comparative results on the MSR-Action3D dataset.

We compare our approach with the state-of-the-art-methods on the cross subject test setting [15]. Results are illustrated in table 3. As the results show, the only approach that outperforms ours is the "Group Sparsity and Geometry Constrain Dictionary Learning" [16]. However, we have to point out that this approach hasn't been extended for real-time action detection yet.

5.5. Role of feature selection in the proposed approach

We put emphasis on the role of the feature selection on the obtained results. We base our analysis on the Video+text modality on MSRC-12 dataset using our 2 levels with overlap configuration and only 200 features are selected out of 10220 features in our descriptor (Table 5 shows the Linear SVM action detection performance for different number of best selected features using SVM-RFE algorithm). The descriptor has 4 concatenated components (1 from level 1 and 3 from level2) each of them has 2555 dimensions[1]. The selected features can be analyzed based on different factors. *First,* every feature is either a mean variable or an entry in the covariance matrix. *Second,* the features are either coming from angles or angular velocities or both. *Third,* using 2 levels and overlapping the features selected might come from different levels and/or different intervals. Analyzing these factors gives justifications and interpretations of our proposed approach.

Table 4 shows the different weights for the aforementioned factors. We can see the mean variables are consistently selected which is an important justification of using the mean variables in our descriptor. There are 102 features selected which is roughly half of the 200 selected features. Whereas only 98 covariance variables are selected from covariance variables in our descriptor. We can see that 149 of the 200 selected variables are from the second level in the pyramid which assures the importance of adding more levels to the temporal pyramid defined in our descriptor.

[1]Out of every 2555 dimensions there are 70 dimensions for the mean vector in that interval and 2485 dimensions for the covariance between the 70 different angles/angular velocities variables

Thirdly an interesting point is that only 46 variables are covariances between angles and angular velocities. This actually means that the more important entries in the covariance matrices in our descriptor are the on-diagonal blocks.

Figure 3 shows that some variables of the 70 angles/angular velocity that we used to design our descriptors happened to occur more frequently in the selected features. Also there are few variables that are never selected. The most selected angles are *(WristLeft, ShoulderCenter, WristRight), (ShoulderRight, ElbowRight, WristRight), (ShoulderLeft, ElbowLeft, WristLeft)*. They occurred 23, 21, and 21 times respectively. The most selected angular velocity *(ElbowRight,WristRight,HandRight), (ShoulderRight, ElbowRight, WristRight), (ElbowLeft, WristLeft ,HandLeft)*. They occurred 16, 15, and 13 times respectively. Actually this shows the importance of the feature selection step in our approach. The more important angles/ angular velocities are repeatedly selected and some other variables are not. This gives insight why the results of the feature selection in table 5 is better than that of our descriptor without feature selection.[2]

Table 1 shows that it is preferable to include more levels in the temporal pyramid. However, computational resources are limiting the real-time performance in that case. The original descriptor size for 3 levels without overlapping intervals will grow up to be 17885 dimensions which will hinder the real time processing. Such descriptor size of 17885 prohibited us from training our classifiers because we had memory limitations with such huge descriptor size. To handle this issue, we used the features selected on the two levels so that we can generalize the selection on the descriptor of 3 levels. The 200 selected features are repeatedly chosen on all sub-intervals of the descriptor which reduced the total size to 1400 features when we use 3 levels without overlapping. As expected the three levels produced the best results and this can be seen in Table 1.

5.6. Real-time Operation

There are two main factors affecting the running time for our online human action detection framework: computing the proposed descriptor and then using the SVM classifier for predicting the performed action in the video sequence. Training the SVM classifier and selecting the most informative features steps are done offline, thus, they don't contribute in the computation latency.

Table 6 shows the average processing time of the proposed detection system for different number of selected features using SVM-RFE algorithm. This is reported on the video+text modality with 2 Levels and overlapping. Advantages of using the feature selection are clear. The speed up is about 5 times when compared to the original descriptor without feature selection. Moreover the descriptor size which is originally 10220 dimensions is effectively reduced. The machine used to generate the table has memory of 16 GB 1600 MHz DDR3 and 2.8 GHz Intel quad-core Core i7 processor.

[2]Sample file for the 200 selected features can be found in the supplemental material with the angles/angular velocity description

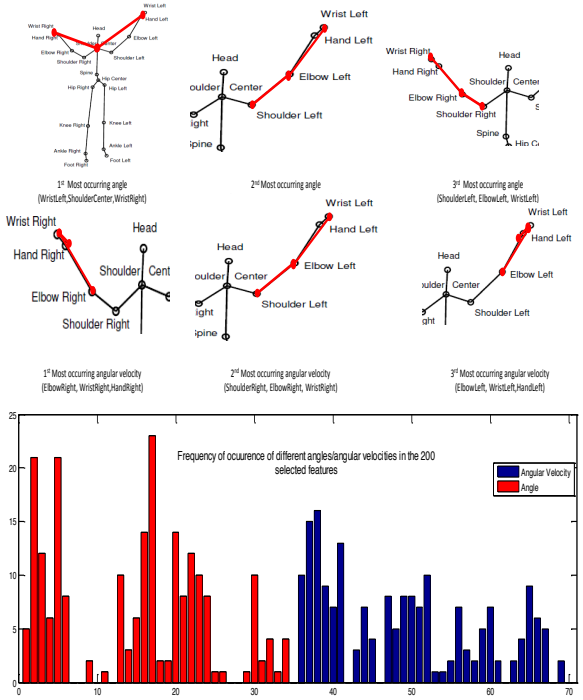

Figure 3. Top: The most selected angles and angular velocities. Bottom: Frequency of occurrence of different angles/angular velocities in the 200 selected features. Zoom is needed to best view the figure.

Feature Source	Selected Features	Percentage
Mean	102	51%
Covariance	98	49%
Angles only	83	41.5%
Angular Velocity only	71	35.5%
Angles and Angular Velocity	46	23%
Level 1	51	25.5%
Level 2, Subinterval 1	54	27%
Level2, Subinterval 2	44	22%
Level 2, Subinterval 3	51	25.5%

Table 4. The 200 Selected features are distributed according to their source from our descriptor.

Features	100	200	300	All Features
Text	0.535	0.547	0.525	0.521
Image	0.601	0.582	0.576	0.540
Video	0.648	0.652	0.664	0.622
Image + Text	0.623	0.648	0.630	0.622
Video + Text	0.694	0.686	0.702	0.705

Table 5. Linear SVM action detection performance for different number of best selected features using SVM-RFE algorithm

Selected Features	average processing time (ms)
100	2.63
200	2.704
300	2.779
All Features (10220)	11.908

Table 6. Average processing time per frame in milliseconds of the proposed detection system for different number of selected features using SVM-RFE algorithm. Video+Text Modality , 2 Levels and Overlap

6. Conclusion

The main contribution of our work is a new approach for real-time multi-scale action detection using a descriptor derived from angles and angular velocities of the 3D joint data extracted from depth sensors. A multi-scale action detection approach is introduced to accommodate for variations in action scales. To achieve real-time performance we used feature selection to reduce the dimensionality of our proposed descriptor. Experiments showed that the accuracy of our descriptor outperformed state-of-the-art methods for real-time action detection [10]. Furthermore an effective feature selection algorithm was applied to reduce the feature size which had a great impact on the computational latency while maintaining or even improving the reported results.

References

[1] J. Aggarwal and M. S. Ryoo. Human activity analysis: A review. *ACM Computing Surveys (CSUR)*, 43(3):16, 2011.

[2] V. Bloom, V. Argyriou, and D. Makris. Dynamic feature selection for online action recognition. In *Human Behavior Understanding*, pages 64–76. Springer, 2013.

[3] V. Bloom, D. Makris, and V. Argyriou. G3d: A gaming action dataset and real time action recognition evaluation framework. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, pages 7–12. IEEE, 2012.

[4] L. Breiman. *Classification and regression trees*. CRC press, 1993.

[5] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[6] 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*, 41(3):786–794, 2014.

[7] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

[8] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[9] K.-B. Duan, J. C. Rajapakse, H. Wang, and F. Azuaje. Multiple svm-rfe for gene selection in cancer classification with expression data. *NanoBioscience, IEEE Transactions on*, 4(3):228–234, 2005.

[10] S. Fothergill, H. M. Mentis, P. Kohli, and S. Nowozin. Instructing people for training gestural interactive systems. In J. A. Konstan, E. H. Chi, and K. Höök, editors, *CHI*, pages 1737–1746. ACM, 2012.

[11] M. A. Gowayyed, M. Torki, M. E. Hussein, and M. El-Saban. Histogram of oriented displacements (hod): describing trajectories of human joints for action recognition. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 1351–1357. AAAI Press, 2013.

[12] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182, 2003.

[13] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1-3):389–422, 2002.

[14] M. E. Hussein, M. Torki, M. A. Gowayyed, and M. El-Saban. Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, 2013.

[15] 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. IEEE, 2010.

[16] J. Luo, W. Wang, and H. Qi. Group sparsity and geometry constrained dictionary learning for action recognition from depth maps.

[17] J. Luo, W. Wang, and H. Qi. Group sparsity and geometry constrained dictionary learning for action recognition from depth maps. 2013.

[18] S. Nowozin and J. Shotton. Action points: A representation for low-latency online human action recognition. Technical report, Technical report, 2012.

[19] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy. Sequence of the most informative joints (smij): A new representation for human skeletal action recognition. *Journal of Visual Communication and Image Representation*, 25(1):24–38, 2014.

[20] Y. Saeys, I. Inza, and P. Larrañaga. A review of feature selection techniques in bioinformatics. *bioinformatics*, 23(19):2507–2517, 2007.

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

[22] Y. Tang, Y.-Q. Zhang, and Z. Huang. Development of two-stage svm-rfe gene selection strategy for microarray expression data analysis. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 4(3):365–381, 2007.

[23] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu. Robust 3d action recognition with random occupancy patterns. In *Computer Vision–ECCV 2012*, pages 872–885. Springer, 2012.

[24] 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*, pages 1290–1297. IEEE, 2012.

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

[26] M. Zanfir, M. Leordeanu, and C. Sminchisescu. The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[27] X. Zhao, X. Li, C. Pang, X. Zhu, and Q. Z. Sheng. Online human gesture recognition from motion data streams. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 23–32. ACM, 2013.