Real-Time Action Detection and Analysis in Fencing Footwork

Filip Malawski, Bogdan Kwolek
Department of Computer Science
AGH University of Science and Technology
Krakow, Poland
Email: fmal@agh.edu.pl, bkw@agh.edu.pl

***Abstract*—This paper is devoted to real-time analysis of continuous footwork training routine in fencing. We propose a model-based adaptive filtering algorithm for accurate selection of segments of interest from a velocity signal acquired by the Kinect motion sensor. We remove false positives from the selected segments by extracting dedicated features and applying a SVM classifier. Finally, we compute parameters of the identified lunge actions, which constitute a feedback for the fencers. The proposed methods are evaluated on a dedicated dataset consisting of actions of eight fencers.**

***Keywords*—real-time action recognition; depth maps; filtering; segmentation; signal processing;**

I. Introduction

Improving skills in sports requires not only motivation and hard work of athletes, but efficient training methods as well. Current technology supports analysis of actions in many sport disciplines. Information gathered with various sensors can provide a valuable feedback for both coaches and athletes. Methods proposed in [1] allow for recognition of karate techniques using the Kinect depth sensor. Detection of swimming strokes with accelerometers is discussed in [2]. Stereo camera and inertial sensors are employed in [3] for golf swing motion tracking.

A common approach to analysis of sports actions is to work on pre-segmented data, where each action is a separate sample [1], [3], [4]. Several state-of-the-art general action recognition algorithms [5], [6] were developed and evaluated on datasets with pre-segmented actions [7], [8]. Detection of selected actions in a continuous sequence is more challenging and rarely discussed in the literature. In sports, analysis of continuous movement has been recently addressed in [2], [9] for detection of cyclic events, namely swimming strokes and pommel horse circles, respectively.

In this paper, we address the problem of detection and real-time analysis of specific actions in fencing footwork training, which is a continuous and non-cyclic movement. In a typical scenario fencers move in a sideway position (see Fig. 1 left). Basic footwork in fencing includes forward and backward steps as well as lunges (see Fig. 1 right), which are used in offensive actions. A proper performance of the lunge action is one of crucial skills for a fencer. Therefore, in this work we focus on detection and analysis of the lunge actions.

For the analysis of fencing footwork we employ the Kinect sensor. It provides both RGB and depth data, as well as automatically detects persons and fits a skeleton model. In our setup the Kinect sensor is located approximately 3 meters from the observed person. The footwork training routine for a fencer includes moving forward and backward with steps and performing the lunge action every few steps. In this paper, we propose methods to detect time segments with potential lunge, verify if the selected time segment is an actual lunge or not, and finally analyze the performed lunge (see Fig. 2).

Detection of lunge candidates is achieved by analysis of the fencer‘s velocity, using the skeleton data provided by the Kinect. Since the Kinect data are noisy and performance of the lunge action differs between fencers, we propose a model-based adaptive filtering method, which allows to accurately extract the interesting time segments. Verification of the candidates includes extraction of specifically designed features, based on the skeleton data, as well as classification of the segments on the basis of the Support Vector Machine (SVM). Analysis of the lunge performance includes computation of various parameters, such as lunge length or acceleration, by employing the skeleton data.

The methods proposed in this paper are verified on a dedicated dataset, consisting of recordings of continuous fencing footwork of eight fencers. A feedback from fencing coaches is provided with respect to the usefulness of the proposed system.

Fig. 1. Fencing position (left) and lunge (right).

This work was partially supported by Polish National Science Center (NCN) under a research grant 2014/15/B/ST6/02808 and dean's grant. The dataset used in this work was acquired due to the courtesy of the Aramis Fencing School (aramis.pl).

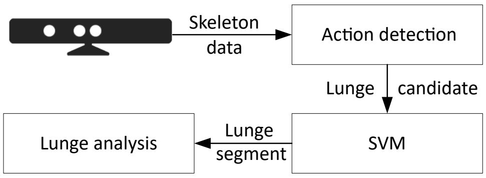

Fig. 2. Scheme of the proposed system.

II. Proposed Methods

A. Detection of the Lunge Candidates

In order to detect the lunge action we need to consider the characteristics of this movement. A lunge is performed by slightly lifting the front leg and then dynamically pushing off the back leg. During the first stage the fencer accelerates rapidly, then the front leg touches the ground and the fencer slows down and finally comes to a short halt. The return to the basic position is done by bending the back leg and moving the front leg backwards. Therefore, the plots of the horizontal velocity over time of a fencer performing a lunge usually have two peaks. The first peak is a maximum, that indicates the moment at which the front leg touches the ground. The second peak is a minimum, that indicates the return to the basic position. Based on these observations, it is evident that detecting lunge candidates can be achieved by finding a sequence of two peaks – a maximum and the following minimum.

Although this criteria is not sufficient, we can easily limit the number of lunge candidates by including another evidence – the maxima must be greater than zero, as the fencer is moving forwards in the first phase of the lunge, and the minima must be lower than zero, as the fencer is moving backwards during the return phase. There are actually only two actions, which meet these criteria – one is the lunge action and the other is a sequence consisting of a step forward and a step backward (see Fig. 3). The proposed method for distinguishing between these two actions is described in the next subsection.

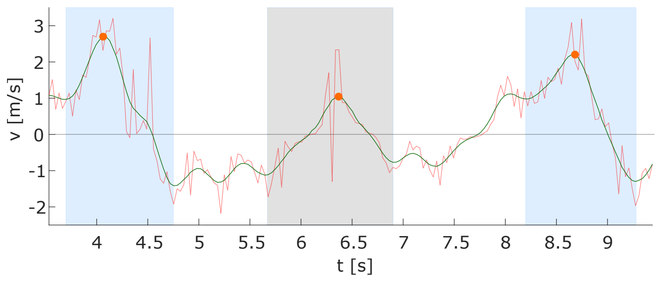

Fig. 3. Example plot of fencer‘s velocity over time. The red line represents raw signal and the green line represents the filtered values. Detection of maximum greater than zero (red dot) and a following minimum lower than zero provides lunge candidates. Left and right candidates are actual lunges (blue background) and middle candidate is a step forward and backward action (grey background).

We propose a method for estimating the velocity of the fencer by employing the skeleton data provided by the Kinect sensor, which contains the positions of 20 joints. In order to measure the velocity of a fencer we consider a joint that is located close to the center of the body, namely the spine base joint [10]. Velocity is determined as the difference of the joint positions in the consecutive frames. Since the Kinect data are noisy, we need to apply a filtering in order to obtain a smooth signal, which would allow analysis of the peaks. Our aim is to provide feedback for the athletes in real-time, therefore we considered filters, which provide local smoothing, with relatively low filter delay. We considered the following filters:

- Moving Average (MA), which computes a simple arithmetic average of velocities v_i over a moving window of length n:

$$ma = \frac{v_1 + v_2 + ... + v_n}{n} \tag{1}$$

  A single smoothing with moving average proved to be insufficient, however, double smoothing, with shorter window in the second iteration provided much better results. Therefore, when referring to the moving average filter we will consider a filter with two iterations.

- Locally Weighted Scatterplot Smooth (Loess) [11], which is computed in the following steps:

  First, given a window of length n, weights w_i are extracted on the basis of the following equation:

$$w_i = (1 - |d_i|^3)^3, \quad i = 1...n \tag{2}$$

  where d_i is the distance between the data point located in the center of the window and the i-th data point, divided by half of the window size.

  Next, a second degree polynomial is fitted using a weighted linear least-squares regression, which minimizes the fit error s given as:

$$s = \sum_{i=1}^{n} w_i (v_i - \hat{v}_i)^2. \tag{3}$$

  where $\hat{v}_i$ is the fitted value.

  The smoothed value of the signal velocity is the value of the fitted polynomial at the center of the window.

In order to properly detect and analyze the lunge actions we need the filters that (i) preserve peaks and (ii) handle outliers. Preserving the peaks is crucial for detecting the exact start of the lunge, which is very important for the further analysis. When the fencer performs lunge after step forward, he/she slows down only temporarily, which generates a small, easy to miss minimum. On the other hand, the Kinect skeleton tracking is unstable, and sometimes the tracked joints positions ‘jump‘ between consecutive frames and thus produce outliers in the velocity data, which can be misidentified as peaks.

Using multiple window sizes we verified how both filters (MA and Loess) were able to preserve peaks and handle outliers. Generally, MA was better suited for the outliers, while Loess was better at preserving the peaks, mainly due to employing the second order polynomial. There is actually a robust version of the Loess filter, dedicated to remove outliers [11], although we found that it works well only when provided with relatively large parts of the signal and is not well suited for real-time signal smoothing. Therefore, we consider only the basic Loess filter.

Fig. 4 illustrates difficulties arising during selecting proper filter type. We can observe that averaging can cause missing the lunge start moment, and preserving the peaks can generate an incorrect peak in the decelerating stage of the lunge. In order to address such problems we propose a model-based adaptive filtering, which allows to combine and take advantage of both filtering methods.

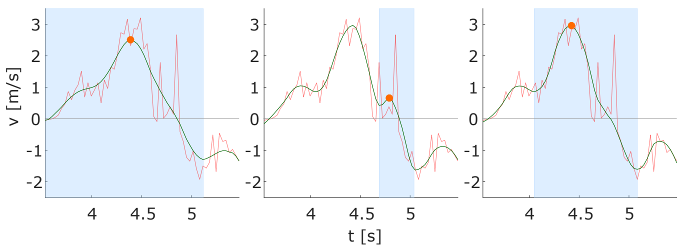

Fig. 4. Effects of different filters on the same signal (velocity over time) with lunge. The red line represents raw signal and the green line represents the filtered values. Left: Moving Average filter, which does not preserve the peak on the rising slope; Middle: Loess filter, which produces an incorrect peak on the falling slope due to an outlier; Right: proposed method.

The main idea is to employ a model of the analyzed movement and based on that knowledge to adapt the filter during the smoothing in order to both preserve the peaks and remove the outliers. In our model, we assume that we need to preserve the peaks when the signal (velocity) is rising and remove the outliers when the signal is falling. Adaptation is achieved by combining filtered values from both filters, according to the current direction (rising/falling) of the signal.

We determine the slope direction parameter dp by taking the mean value of the first derivative of the filtered signal in a window of size equal to the filter window size:

$$dp = \sum_{i=1}^{n-1} \frac{\hat{v}_{i+1} - \hat{v}_i}{n-1}. \tag{4}$$

where n equals to window size.
The direction parameter dp is then cut to range (-1, 1):

$$dp = Max(Min(dp, 1), -1) \tag{5}$$

and normalized to range (0,1):

$$dp = \frac{dp + 1}{2} \tag{6}$$

Therefore, when $dp = 1$ the slope is rising with 45 (or more) degrees, and when $dp = 0$ the slope is falling with 45 (or more) degrees. We can now use the direction parameter directly for the weights for combining the filters:

$$w_{movingavg} = 1 - dp \tag{7}$$

$$w_{loess} = dp \tag{8}$$

It is worth mentioning, that the filtered slope does not change rapidly, therefore the transition between the two filters is smooth as well. By combining the filters we can achieve proper filtering of the velocity signal (see Fig. 4 right).

B. Classification of the Lunge Candidates

As discussed in the previous section and presented in Fig. 3, we need to verify the lunge candidates by distinguishing between the lunge action and an action consisting of a step forward and a step backward. Although the horizontal velocity pattern of the spine base joint is similar in both cases, the overall motion is quite different. Therefore, we decided to employ features based on the velocities of all joints.

The feature extraction scheme is as follows. Once the lunge candidate segment is found we compute velocities (both horizontal and vertical) for each joint. We then interpolate the signal to a common length. Since the average duration of an extracted lunge segment is approximately 1 sec and the Kinect operates with 30 Hz, we interpolate the signal to 32 sample points. We then transform the signal to the frequency domain by applying the Fast Fourier Transform (FFT). By taking first 3 coefficients of the FFT for each of the 20 joints, in both x and y directions, we obtain a feature vector with 120 features.

Given a feature vector with a constant length we can apply the Support Vector Machine (SVM) classifier. We consider both a linear SVM as well as a non-linear, with Radial Basis Function (RBF) kernel.

C. Lunge analysis

Once the lunge segment is identified we can analyze the motion of this action. Based on consultations with fencing coaches we analyze the following parameters:

- Hand delay - in offensive actions it is crucial to first straighten the armed hand and then move forward with the lunge. Therefore we compute the time difference between straightening the arm and the beginning of moving forward. We consider the arm to be straighten when the angle formed by the joints: shoulder, elbow and wrist is higher than 160 degrees. This parameter requires determining the exact start of the lunge.
- Lunge length - computed as the difference between the extreme positions of the spine base joint during the lunge.
- Acceleration - computed as the average change of velocity of the rising slope.
- Time - from the start of the lunge to the end of the returning phase (determined by the minimum peak).

During training the fencer can decide to improve any of these parameters in his/her lunge action performance. Professional fencers may decide to work on multiple parameters simultaneously, which is more difficult, as some of them are mutually dependent (e.g. longer lunge requires more time).

III. Experiments

A. Dataset

For evaluation of the proposed methods we recorded a dedicated dataset, which includes data from eight fencers, with different skills level (from intermediate to advanced). In each recorded sample a fencer was asked to perform the footwork training routine, including approximately five lunge actions. For each fencer we collected from two to seven samples. In some cases the recordings were done in two sessions (on different days). We acquired totally 31 samples, which include total of 149 lunge actions. The segments with lunge actions were manually labeled in each recording and used as ground-truth in the evaluation.

B. Detection

In order to verify the lunge detection we compared the lunge segments found by the proposed method with the manually labeled segments. We considered two parameters: (i) how many of the lunges were found, (ii) how accurately was the start of the lunge determined. Here we did not consider falsely detected lunges (this will be discussed in the next subsection). A lunge was considered as found when at least 50% of the manually labeled segment was determined as the lunge segment by the algorithm. The accuracy of finding the lunge start frame was computed as difference between the manually labeled and the detected starting points.

Table I presents results for the Moving Average, Loess and the proposed model-based adaptive filter. As we can observe, our method is superior both in terms of finding the lunges and minimizing the error of finding the starting point. It allows to find all lunge segments, with average starting point error equal to 1.91 frame, which corresponds to 63 ms. Such accuracy is sufficient in this scenario.

TABLE I. Results of the lunge segments detection, using different filters

Filter	Window(s) size	Missed lunge segments	Avg (with std dev) starting point err. (given in frames)
Moving Average	7; 5	5	3.65 ± 4.43
Moving Average	9; 7	7	5.63 ± 6.04
Loess	15	11	3.14 ± 4.74
Loess	17	4	2.50 ± 3.69
Proposed method	15	0	1.91 ± 1.82
Proposed method	17	0	2.15 ± 2.27

C. Classification and Analysis

Experiments with the lunge segment candidates classification were conducted using leave-one-out cross-validation. Having on regard that the dataset consisted of eight persons we performed eight folds cross-validation – in each fold one person was used as the test set and the other ones as the training set. The dataset consisted of total of 149 actual lunges. In all folds the proposed filtering method (with window size set to 15) produced 205 lunge segment candidates. Both linear and RBF kernel SVM were used and a grid search for the best parameters was conducted. In both cases the classifier correctly recognized all but one sample, resulting in 99.51% accuracy.

The parameters of the proposed lunge analysis were discussed with fencing coaches, who found them to be sufficiently accurate to provide a useful feedback.

IV. Conclusions

In this paper we addressed the problem of real-time analysis of continuous, non-cyclic movement in the fencing footwork training routine. We developed a method for finding the action of interest (lunge), by analyzing the fencer‘s velocity based on the Kinect skeleton data. We proposed a model-based adaptive filtering algorithm, which allows to find the lunge segment candidates accurately in real-time. Applying an SVM with the proposed features allowed to remove almost all false candidates. Finally, we investigated a useful analysis of the lunge action, by computing a set of dedicated parameters.

Based on the opinions gathered from the fencing coaches, the discussed methods can constitute a valuable tool in fencing training for two reasons. First, it can provide feedback for a fencer without supervision from the coach. Secondly, providing numerical values for the lunge performance parameters is a motivational factor, as the fencers can track their progress and compete with others.

References

[1] T. Hachaj, M. R. Ogiela, and K. Koptyra, “Effectiveness comparison of Kinect and Kinect 2 for recognition of Oyama karate techniques,” in *Network-Based Information Systems (NBiS), 2015 18th Int. Conf. on*. IEEE, 2015, pp. 332–337.

[2] M.-S. Pan, K.-C. Huang, T.-H. Lu, and Z.-Y. Lin, “Using accelerometer for counting and identifying swimming strokes,” *Pervasive and Mobile Computing*, vol. 31, pp. 37–49, 2016.

[3] C. N. Nam, H. J. Kang, and Y. S. Suh, “Golf swing motion tracking using inertial sensors and a stereo camera,” *IEEE Trans. Instrum. Meas.*, vol. 63, no. 4, pp. 943–952, 2014.

[4] F. Malawski and B. Kwolek, “Classification of basic footwork in fencing using accelerometer,” in *Signal Processing: Algorithms, Architectures, Arrangements, and Applicat. (SPA), 2016*. IEEE, 2016, pp. 51–55.

[5] O. Oreifej and Z. Liu, “HON4D: histogram of oriented 4d normals for activity recognition from depth sequences,” in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recogn.*, 2013, pp. 716–723.

[6] X. Yang and Y. Tian, “Effective 3d action recognition using eigenjoints,” *J. of Visual Commun. and Image Representation*, vol. 25, no. 1, pp. 2–11, 2014.

[7] L. Zhang, J.-C. Hsieh, and J. Wang, “A Kinect-based golf swing classification system using hmm and neuro-fuzzy,” in *Int. Conf. on Computer Science and Information Processing (CSIP)*. IEEE, 2012, pp. 1163–1166.

[8] J. Wang, Z. Liu, Y. Wu, and J. Yuan, “Mining actionlet ensemble for action recognition with depth cameras,” in *IEEE Conf. on Computer Vision and Pattern Recogn. (CVPR)*. IEEE, 2012, pp. 1290–1297.

[9] B. Reily, H. Zhang, and W. Hoff, “Real-time gymnast detection and performance analysis with a portable 3d camera,” *Computer Vision and Image Understanding*, 2016.

[10] MSDN. Kinect joint enumeration. [Online]. Available: https://msdn.microsoft.com/en-us/library/microsoft.kinect.jointtype.aspx

[11] W. S. Cleveland, “Robust locally weighted regression and smoothing scatterplots,” *J. of the Amer. Stat. Assoc.*, vol. 74, no. 368, pp. 829–836, 1979.