AGH University of Science and Technology

Faculty of Computer Science, Electronics and Telecommunications

Automatic analysis of techniques and body motion patterns in sport

Ph.D. Dissertation

Author:
Filip Malawski

Supervisor:
Prof. Bogdan Kwolek

Kraków, 2019

Akademia Górniczo-Hutnicza im. Stanisława Staszica

Wydział Informatyki, Elektroniki i Telekomunikacji

Automatyczna analiza technik oraz form ruchu ciała w sporcie

Rozprawa doktorska

Autor:
Filip Malawski

Promotor:
dr hab. inż. Bogdan Kwolek, prof. AGH

Kraków, 2019

Acknowledgments

Firstly, I would like to express my sincere gratitude to my thesis supervisor, Professor Bogdan Kwolek, for his support and assistance in carrying out the research described in this dissertation.

I would also like to thank Professor Krzysztof Zieliński, my first mentor during my doctoral studies, as well as my colleagues from the Computer Science Department, for their support and collaboration on multiple projects.

My gratitude goes also to the fencers and coaches from Aramis Fencing School for sharing their time and knowledge. Their participation was crucial for conducting and evaluating this research.

A special gratitude goes to my wife, Kamila, for her patience, constant support and providing invaluable help whenever it was needed.

For my family and friends, I thank you all, for always being there for me.

I also acknowledge that this work was supported by Polish National Science Center under research grants 2014/15/B/ST6/02808 and 2016/21/N/ST6/00553, Polish National Center for Research and Development - Applied Research Program under grant PBS2/B3/21/2013, and AGH Faculty of Computer Science, Electronics and Telecommunications dean's grants 15.11.230.280, 15.11.230.322 and 15.11.230.398.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

This chapter presents the context of the research conducted in this work. First, the background for the subject of the thesis and the motivation for addressing the specific issues are discussed. Then, the research goals and contributions are presented. Finally, the chapter structure of the thesis is shown.

The use of modern technology is a part of many sports disciplines, both during competitions and trainings, in various aspects. Measurements in sports became automated and more precise due to the development of electronic sensors [205]. Laser and video sensors allow to measure times of sprinters with high precision [118]. RFID sensors are used in long distance runs to automatically detect each person crossing the finish line [118]. Video recordings are used in ball games, such as volleyball or tennis, to verify whether the ball hit the ground inside the court [83, 106]. Electric scoring devices are used in fencing for hit detection [117].

In recent years, personal devices such as smartphones, smartwatches or fitness trackers have begun to be used more and more widely in sports training. Based on signals from the GPS and inertial sensors, as well as pulsometers, these devices are capable of providing various types of useful information. Route recording is available for runners and cyclists, and includes time, distance, and speed statistics [252]. The runners can optimize their pace based on heart rate monitoring [232]. Step counting, based on the accelerometer signal, provides feedback on the amount of everyday activities [53]. Dedicated smartphone applications allow to track progress during training, by recording the number of repetitions performed during each training session [71].

The discussed applications, while being very useful, seem to be limited, considering the current state-of-the-art methods used in signal processing and machine learning. Computer vision algorithms for the recognition of general actions have been investigated for many years, and are capable of providing high accuracy under various conditions [312]. Many successful studies on specific applications, such as gait recognition [148] or fall detection [145], have been reported as well. Pedestrian detection is already being used in commercial driver assistance systems [89]. Inertial sensors are being employed for gesture and sign language recognition [297]. Despite the richness of the available methods for the analysis of human motion, few of them have found a way to be applied in sports, thus motion analysis in sports poses an interesting research topic.

1.1 Sports motion analysis

Quite limited application of general human motion analysis methods in sports stems from certain characteristics of athletes' motion and the expected feedback. First of all, actions in sports differ from actions performed in typical everyday activities, therefore models of generic actions are not directly applicable. Secondly, motion in sports is usually faster and more dynamic, which makes the tracking more difficult. Sports actions also require a lot of precision, therefore motion analysis methods need to be precise as well in order to provide useful feedback. Quantitative information, such as how many actions of a given type were performed, is often not sufficient. What is more needed, it is qualitative information providing feedback on how well an action was performed, as that is the basis for improving sports skills.

Another issue is the differentiation of actions between various sports. Usually, motion analysis methods which are suitable for one sport are not directly applicable to others. Rather than developing general methods for all sports, researchers have to address each sport separately in order to provide precise feedback. On the other hand, methods from one sport can often be adapted to others, as long as some similarity of motions exists.

In this work, both techniques corresponding to actions, as well as body motion patterns corresponding to sequences of actions, are considered. Temporal segmentation of body motion patterns is needed to extract time segments containing the executions of specific techniques. Then, the detected actions can be analyzed in order to provide qualitative feedback, which is useful for athletes and coaches.

1.2 Thesis statement

The goal of this thesis was to develop methods which would support improving sports skills. This objective is attained by employing the analysis of techniques and body motion patterns for providing the qualitative measurements essential for this task. In the proposed approach, the extracted information can be used both by athletes and coaches, during or after practice. The thesis of this dissertation is formulated as follows:

> *Automatic analysis of techniques and body motion patterns of athletes generates feedback, which is useful for improving sports skills*

The dissertation focuses on several aspects, which are related to providing feedback relevant for improving sports skills. First of all, the problem of distinguishing specific sports actions is addressed. Contrary to general action recognition, in sports, different dynamics of motion can result in different actions. Secondly, the temporal segmentation of continuous training routines is considered, as this is necessary to perform qualitative analysis. Typically, this stage is omitted in the literature, as most algorithms are verified on pre-segmented data. Thirdly, methods for computing performance parameters are proposed as well. Finally, efficient presentation of feedback is addressed. The solutions proposed in regard to all of the discussed issues can operate in real-time, which is a crucial aspect in terms of usability. The research is conducted on a single sports discipline, based on the aforementioned assumption, that the methods developed for one sport need, and can be adapted to others.

1.3 Research contribution

This thesis considers automatic analysis of motion in fencing, which was chosen as the sports discipline of interest, due to several identified research challenges (see Section 2.3.5). The main contributions of this dissertation are as follows:

- a comprehensive review of state-of-the-art methods for human motion analysis, and sports analysis in particular, which are based on various signal modalities
- novel methods for the classification of similar sports actions which are based on motion dynamics, including several novel feature extraction algorithms, employing both visual and inertial data, as well as new feature selection and fusion methods
- a dedicated dataset with fencing footwork actions, used to verify the classification methods, which was made publicly available
- evaluation of the proposed classification methods on two datasets including a publicly available UTD-MHAD dataset; the proposed methods are shown to outperform state-of-the-art algorithms for both datasets
- novel methods for the detection of actions in a continuous fencing footwork training routine, as well as for the qualitative analysis of the extracted segments, which are based on both visual and inertial signals
- a proof-of-concept implementation of a system capable of detecting and analyzing actions in a fencing footwork training routine in real-time
- evaluation of the proposed footwork action detection and analysis methods by using two dedicated datasets
- novel methods for blade tracking, weapon action model learning, bladework practice evaluation, and providing immersive feedback by using augmented reality
- a proof-of-concept implementation of an augmented reality-based system for supporting bladework practice
- evaluation of the proposed methods for bladework practice support

1.4 Dissertation structure

The structure of the dissertation is as follows. Chapter 2 presents the background for the research. A thorough review of human motion analysis methods is presented. Then, sports motion analysis is discussed and the discipline of interest is introduced. Chapter 3 addresses the problem of fencing lunge action classification by employing motion dynamics. A dedicated dataset is introduced, methods for feature extraction, selection, and fusion are presented, and an extensive evaluation is provided. In Chapter 4, methods for the detection and analysis of actions in a continuous fencing footwork routine are proposed. Their evaluation is based on two dedicated datasets which allow for comparisons between the results obtained in regard to visual and inertial data. Chapter 5 focuses on providing real-time, immersive feedback by employing augmented reality. Issues regarding blade tracking, weapon action model learning and creating mixed virtual-real views with augmented reality glasses are addressed. Chapter 6 concludes the dissertation and presents directions for further work.

Chapter 2

Background and related work

This chapter discusses context of the research in detail and explains the motivation behind addressing the specific problems of the field. First, the problem of motion analysis in sport is introduced. Then, a thorough review of methods for automatic human motion analysis is conducted. Finally, automatic motion analysis in sport is discussed, and the research objectives are presented.

Sport plays an important role in modern society [51]. It allows for social interaction not only for athletes during training and when taking part in competitions, but for sports fans as well, due to attending sports events and emotional engagement involved [122]. It is greatly beneficial for health, both in terms of one's physical [284] and mental [84] well-being. World-class sports competitions, such as the Olympic games, are a significant part of global culture. Sport is commonly present in our live, and it is therefore justified that many researchers make efforts to help understand and positively influence the various aspects of this domain.

There are two groups of people that can benefit the most from introducing novel technologies in sports: the athletes and the spectators. In this work, the focus is on developing motion analysis methods, which could provide useful feedback for the athletes. This includes precise tracking of the performed movement, the temporal segmentation, detection, and recognition of specific actions, the qualitative analysis of these actions, and, finally, providing useful feedback, preferably in real-time. Issues regarding the presentation of sports activities to the spectators, for instance by the detection of the key events of a sports broadcast [99], are not a part of this research and are therefore discussed only briefly.

This chapter is organized as follows. In Section 2.1, the concept and the goals of motion analysis in sports are presented, as context for the research. Section 2.2 includes a detailed review of the methods used in the field of automatic human motion analysis which constitute a reference point for developing more specific methods for sports analysis. In Section 2.3, related work is presented, new challenges are identified, and the discipline of interest - fencing - is introduced. Fencing is a highly dynamic and technical sports discipline, in which the identified challenges are relevant. Finally, the research objectives are presented and the chapter is summarized.

2.1 Motion analysis in sports

From an athlete's point of view, one of the main objectives of practicing a sport is to develop the proper skills and achieve the best possible results in sports competitions. Although the amount of effort put into the task is very important, little can be accomplished without the proper training methods. It is essential to assess performance and provide feedback to improve skills in sports. The motivation behind this work is that both performance assessment and feedback can be effectively supported by automatic motion analysis.

2.1.1 Sports skills

Sports can be perceived as a task of employing skills in a competitive activity with well-defined rules. In order to achieve better results, extensive practice is required, as it enables the improvement of skills. In order to understand how the process of increasing one's skill level works, we first need to define skill. Stallings [249] distinguishes three domains of skill: cognitive (knowledge of what to do and how to do it), perceptual (interpretation of information from the surrounding environment), and motor (performing movement). A skill is always composed of all three domains, although we can usually distinguish a dominant domain for a particular task. In sports analysis, motor skills are considered, which are defined by Stallings as follows:

> *A motor skill is a learned, goal-directed activity accomplished primarily through muscular contributions to action and entailing a broad range of human behaviors.*

Stallings emphasizes that this definition highlights an important feature of motor skills - that they can be improved by learning. A skill is influenced by two factors - abilities and technique, both of which are important in the learning process [218]. Abilities such as strength, flexibility, balance, and coordination [249] define the maximum possible characteristics of a performed movement, such as range, speed or accuracy. Technique is a motor procedure for tackling a motor task [218]. It defines how a certain action should be performed, for instance, how to hit the ball in the context of a serve in volleyball or how to perform a parry in fencing. An athlete becomes skillful by learning how to properly execute a particular technique, which requires certain abilities. While technique is the basis for movement in sports, the body motion pattern is a sequence of techniques performed by an athlete. For instance, the body motion pattern in a triple jump includes techniques such as running, hopping, stepping and jumping.

The analysis of techniques and body motion patterns is essential for coaching in sports for two reasons. First of all, it allows to provide feedback to athletes, which is necessary in order for them to correct their movement, and eventually improve their skills. Secondly, the technique itself may be improved - the coach may decide that a certain movement should be performed differently. This thesis focuses on providing feedback to athletes, although some of the presented methods may also be employed for analyzing techniques in general.

2.1.2 Performance assessment

Typically, athletes can assess their performance through proprioception, measurements and feedback from a coach. Proprioception is the sense of the relative position of one's

own body parts and the amount of effort being exerted in a movement [237]. Unfortunately, proprioception is rather limited and not very reliable - our perception of our own motion often differs in reality. Measurements, on the other hand, can provide a highly accurate assessment of a performed action. For instance, a sprinter can measure their time in a 100-meter dash, and a jumper can measure the length of their jump. However, when the assessment of the performance of a particular technique is needed, rather than just the final outcome of a sports action, measurements are difficult to make. Nevertheless, as will be shown in this work, by applying sophisticated sensors and data processing algorithms, accurate analysis of techniques and body motion patterns in sport is possible. The third type of feedback, provided by a coach, is usually the most valuable for an athlete. The coach knows how the technique should be performed, observes the errors that are being made and informs on how to correct them. However, the availability of the coach is an issue, as they usually need to share their attention between multiple athletes. Also, quick and fine motions may be difficult to observe, even for an experienced coach.

The goal of this work is to provide athletes with feedback based on advanced measurements and therefore allow them to efficiently improve their skills even in the absence of a coach. An additional objective is to provide coaches with relevant and precise information which would be difficult to obtain with simple observation. It is worth noting that the automatically gathered information could also be used to create repositories of data, for use in statistical analysis and progress tracking. These goals can be realized by the analysis of techniques and body motion patterns. Athletes usually practice techniques separately at first, and then combine them into sequences. Therefore, during an automatic analysis, the observed body motion pattern needs to be segmented in time in order to detect particular techniques. An athlete executes techniques with regard to their level of proficiency as well as their personal style, therefore the models of techniques need to account for such variations.

For the sake of completeness of the argument we need to address another aspect of an athlete's preparation, which is tactics [111]. While techniques define how to perform specific actions, tactics regards knowing when and how to employ a technique. Tactical skills are particularly important in sports where direct interaction with an opponent is present. Although interesting and important, the analysis of tactics is outside the scope of this work, as the focus is placed on the techniques. It is, however, worth noting that the ability to recognize specific techniques is crucial for analyzing tactics, and therefore the presented methods are useful also in this context.

One of the key characteristics of analyzing motion in sports is that each sport has its own unique set of techniques and body motion patterns. This makes it very difficult to develop generic methods for sports action analysis. For instance, tennis players need to perform the movement of hitting a relatively small ball with a racket, while in football, most techniques are limited to the legs. There seems to be very little correspondence in the analysis of these two disciplines. On the other hand, once we are able to reconstruct the motion trajectory of the upper limbs of a tennis player, we can expect to develop similar methods for tracking the lower limbs of football players. Therefore, rather than trying to create general methods that could be employed in any sport, this work focuses on analyzing specific actions in a specific discipline in order to develop methods which could be adapted for other disciplines as well.

2.2 Automatic analysis of human motion

Automatic analysis of human motion can provide information in regard to which actions are being performed and how. There are multiple applications for such methods, including human-computer interaction [167, 183], surveillance [168], gait recognition [142], fall detection [146], facial expression recognition [178], sign language recognition [116, 130, 172] or sport practice support [226]. The most common task is action recognition, for which the goal is to classify actions, usually based on previously gathered knowledge [223]. This is most often done with pre-segmented data, where the actions have already been extracted [34], although in real-time applications, the temporal segmentation of motion remains an important issue. The process of action recognition includes the acquisition of data from a sensor, the extraction of relevant features, and applying machine learning methods to build classifiers capable of distinguishing between actions [250, 287]. Most works in the literature focus on proposing novel features. In regard to machine learning, general methods are typically applied and optionally tuned for specific scenarios. What is more, deep learning methods have recently become popular, where features can be extracted automatically by a neural network which also performs classification [109]. In the qualitative analysis of motion, the goal is to identify and measure the important characteristics of an action. Although the description of motion may be similar to that in the case of action recognition, the difference is that instead of classification, the relevant parameters are extracted in order to analyze the performance of an action. Qualitative analysis of motion is particularly important for providing useful feedback in sports [138].

In this section, a thorough review of the literature in the area of human motion analysis is provided. While most works focus on the action recognition task, the described methods are largely applicable to qualitative analysis as well, since the most important aspect in both cases is description of motion. This section starts with the definition of action, which is a basic term in human motion analysis. Next, the key challenges in the field are presented and an overview of methods found in the literature is provided. The most important methods in the field of automatic human motion analysis are discussed in more detail in separate subsections.

2.2.1 Actions

Automatic analysis of human motion is usually considered in terms of actions and activities. Although an action seems to be an intuitive concept, its exact definition varies between authors. Activity is commonly considered to be a broader term than action, although, the exact boundary between the two concepts is rather vague. Herath et al. [109], in their survey on action recognition, refer to several definitions of both action and activity in the literature. According to [193], an action, for instance running, is a movement composed of action primitives such as moving a leg forward, while an activity is a number of subsequent actions. The authors of [263] characterize actions as simple motion patterns executed by a single person and of a short duration, while defining activity as a complex sequence of actions performed by multiple persons. Wang et al. [283] emphasize the result of an action (in terms of interaction with the environment) to be the defining factor. Based on that, Herath et al. [109] propose their own definition: *'Action is the most elementary human-surrounding interaction with a meaning'*. Gestures are another related concept, commonly associated with hand gestures and sign language recognition in particular. A gesture can be considered to be a type of action, however it

can also be understood as a basic component of an action [157]. In this work, an action is considered to be an execution of a technique, and a sequence of techniques is referred to as a body motion pattern. Other terms, like activities and gestures, are not used.

2.2.2 Challenges

In order to be useful in real-life applications, human motion analysis methods need to be robust to multiple factors which may vary depending on the scenario. There are several challenges which need to be addressed [223, 287]. First of all, variations in the performance of actions always occur. At least slight variations are expected, even when the same person is repeating the same action under the same conditions. Different persons may perform an action in a significantly different manner due to their personal style and anatomical construction. The rate of performance needs to be considered as well, as it may vary depending on the person and the situation. Environmental conditions may also constitute a relevant factor. For instance, running on sand and running on a road will result in different leg movement. Environment is also important in the context of the data acquisition and processing. Video-based methods are often vulnerable to changing lighting conditions, occlusions, and different viewpoints [223]. The difficulty of localizing a person can depend on complexity of the background. These issues can be addressed by applying depth cameras, although they are more expensive and operate in a limited range, and can therefore not always be employed. Inertial sensors are free of the problems specific to visual sensors, but they have their own limitations, such as battery life, data transfer, signal noise and the necessity to wear the sensor. In order to gain robustness, human motion analysis methods should address all of the aforementioned issues.

Most of the works available in the literature focus on general action recognition, which mostly includes significantly differing actions [34]. The recognition of similar actions is rarely addressed, even though it is an important issue particularly in sports motion analysis. It is also worth noting that many works do not consider the task of temporal segmentation, and operate on pre-segmented data instead [309]. The problem of temporal segmentation is relevant in the context of real-world applications. Qualitative analysis of actions is also rarely performed, as most works are limited to the classification task [312]. Finally, in some scenarios, particularly in sports, the issue of providing feedback is crucial, although it is not addressed in the typical case of general action recognition. While it is useful to perform analysis after an action has been finished, e.g. based on a video recording, the real challenge is to provide feedback in real-time.

Data availability is an important aspect as well, as it is necessary for the development and evaluation of different methods [34, 251, 309]. The data acquisition process is usually time-consuming [169]. The key challenge is to record sufficient variations to allow machine learning algorithms to achieve generalization. Public action recognition datasets have become popular, as they allow for reliable comparisons between different methods.

2.2.3 Methods overview

One of the basic factors in the examined works is the type of the analyzed signal. Most commonly employed sensors include: cameras, providing RGB videos; depth sensors, providing depth map sequences; and inertial measurement units (IMU), providing data from accelerometers, magnetometers, and gyroscopes. Motion capture (MoCap) systems

are also used. They employ a number of coordinated IMUs or multiple cameras tracking a set of markers in order to reconstruct human motion with high accuracy. Fusion of multiple modalities is also an interesting research direction.

An overview and discussion of the methods based on RGB data can be found in a number of survey papers [24, 109, 193, 223, 263, 287, 312]. Approaches to the classification of these methods vary between authors, as they highlight different aspects. Reviews devoted to depth and inertial sensors can be found in [45, 157, 225, 300] and [6, 10, 54, 151, 220] respectively. Based on the survey papers and the analyzed literature, the following categories of methods are distinguished and discussed in this work:

- Body-based methods - a person is detected in the scene and either whole figure movement is considered or the body is segmented into body parts, so that their movement can be analyzed separately or with respect to one another
- Spatio-temporal interest points - points distinctive in both space and time are detected, then dedicated descriptors around these points are computed
- Grid-based local descriptors - motion is described in a per-pixel fashion or in cells of a spatio-temporal grid
- Deep learning - action recognition is based on deep neural networks which can operate on raw images without the explicit feature extraction step
- Depth-based methods - depth maps are employed either directly or for skeleton tracking, which enables the analysis of joint motion
- IMU-based methods - data from accelerometers, gyroscopes, and magnetometers are employed
- Multi-sensor systems - multiple homogeneous sensors are used for motion capture systems or heterogeneous sensors are employed for multimodal data fusion

The most important methods in each category are described in the following subsections.

2.2.4 Body-based methods

Body based methods aim at detecting, tracking, and, optionally, segmenting the figure of a person. The figure can be represented as a silhouette or a contour, usually extracted by background subtraction [219]. In [23, 63], silhouettes are used in order to generate motion energy images (MEI) as well as motion history images (MHI) (see Fig. 2.1). MEIs are binary cumulative motion images indicating regions of motion, while in MHI, pixel intensity is a function of the temporal motion at a given point, computed using a decay operator. The MEI and MHI of actions constitute temporal templates which can be matched by computing Hu moments [119]. Weinland et al. [286] extend the temporal templates to 3D by introducing motion history volumes (MHV). 3D representations of silhouettes in time are also employed in [19, 94] to construct space-time shapes, from which features such as space-time saliency and space-time orientations are extracted. The contours of silhouettes are used in [41] for generating so-called star skeletons. Both silhouettes and contours are utilized in [278] for computing average motion energy (AME) and mean motion shape (MMS) representations.

Modeling background for subtraction may be difficult, therefore alternative methods for finding persons in images were proposed. One of the most efficient and popular

approaches to human detection is employing histograms of oriented gradients (HOG) [60]. The authors of [258] extend the HOG descriptor for recognizing pose primitives. A method for both tracking and action recognition using the PCA-HOG descriptor is proposed in [162]. Once a person is detected in an image, it is possible to employ 2D or 3D parametric body models in order to obtain relevant information with respect to motion of specific body parts. In [96], a simple 2D model in the form of a stick figure is fitted to a silhouette. The authors of [197] localize joints and limbs in an image based on multiple visual cues. A method for recovering a 3D human pose from a monocular image is presented in [3]. In [196], 3D human body configuration is estimated by employing a shape context. A detailed survey on pose estimation and modeling is presented in [222].

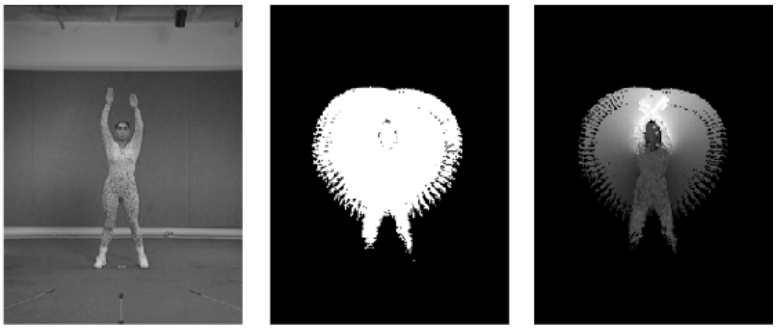

Fig. 2.1: Silhouette-based action recognition - original image (left), motion energy image (center) and motion history image (right) (source: [23]).

2.2.5 Spatio-temporal interest points

Instead of focusing on finding a person in a video, it is possible to track a number of keypoints which are distinctive in both space and time. One of the main advantages of such an approach is the increased robustness to occlusions and changing conditions. Space-time interest points (STIP) were introduced in [149], where the authors extended the Harris corner detector [104] to 3D in order to construct a model of a walking human. Dollar et al. [69] observed that stable interest points are rather rare and addressed this issue by proposing to employ Gabor filtering on spatial and temporal dimensions separately, in order to improve interest point detection. Hessian-based detectors proved to be efficient as well, as reported in [293].

The detected interest points can be used for motion analysis in different ways. One popular approach is to compute local descriptors around the interest points. Spatio-temporal image gradients within Gaussian neighborhoods are used in [238] for human action recognition. Lowe [161] assigned a scale and an orientation to each interest point and used this information for computing gradient-based descriptors in regions around the keypoints (see Fig. 2.2). The proposed approach is called the scale invariant feature transform (SIFT). SIFT features are invariant to both scale and orientation, and enable efficient matching between different views of objects or persons. SIFT proved to be crucial for many applications, such as object recognition [161], robot localization [240] or image stitching [155]. SIFT was later extended to 3D SIFT [239], which is computed in spatio-temporal cubes rather than on single images. The authors of [14] addressed the SIFT feature matching process efficiency problem and proposed speeded-up robust features (SURF), which provide similar recognition accuracy at a lower computational cost. An evaluation of local spatio-temporal features can be found in [82, 273].

Another approach for employing interest points is to analyze their trajectories of motion. In [191], the velocity of the tracked points is computed over a long temporal range, and such sequences are then employed for classification by using the bag-of-words model [246]. The authors of [186] argue that short trajectory snippets, denoted as trajectons, are better suited for action recognition. Dense trajectories and improved dense trajectories, introduced in [271] and [272] respectively, include multiple different descriptors that are computed along densely sampled trajectories. The motion relationships between different objects are considered in [125] by inclusion of local and global reference points for computing trajectory shape descriptors as well as motion representation.

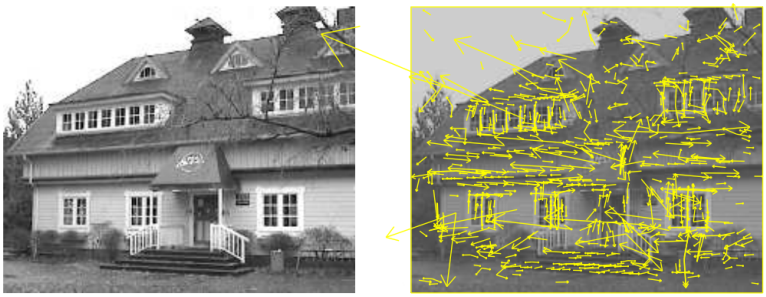

Fig. 2.2: Detection of space-time interest points - original image (left) and the detected keypoints, including location, scale, and orientation (right) (source: [161])

2.2.6 Grid-based local descriptors

While descriptors such as SIFT or SURF were devised specifically to be used with spatio-temporal interest points, there are multiple relevant descriptors proposed in the literature, which do not require finding keypoints. An alternative approach is to divide the spatio-temporal volume with a grid and compute the descriptors in cells [285]. It is worth noting, that such descriptors can also be computed in 3D neighborhoods defined around the interest points [150].

HOG can be used not only for person detection, but for motion description as well. In [137], 3D gradients are employed to construct a spatio-temporal descriptor. HOG 3D is also employed in [285] to deal with occlusions and viewpoint changes. Local binary patterns (LBP) encode gradient changes around a point in a binary manner. This descriptor was primarily used for texture recognition [208], and later extended to three orthogonal planes (LBP-TOP) for the description of dynamic textures in facial expression recognition. LBP-TOP was also employed for action recognition [133]. The shape context [16] was designed to describe mutual relations of points in the contours of shapes, including distance and orientation. It was also adapted for pose [163] and motion description [311].

Optical flow is a well-known algorithm which provides motion information by computing the pixel-wise oriented difference between consecutive frames [15] (see Fig. 2.3). Such information proved to be useful for human motion analysis. The authors of [73] produce four separate channels from the optical flow by taking the x and y components in both positive and negative directions and apply such features for action recognition at a distance. Motion descriptors based on optical flow are also proposed in [62]. The histogram of optical flow (HOF), inspired by the HOG descriptor, was introduced in [150]. Variations of HOF, including motion vector orientations, were presented in [61]

and [36]. The HOF descriptor is also popular in methods combining multiple descriptors, such as dense trajectories [272].

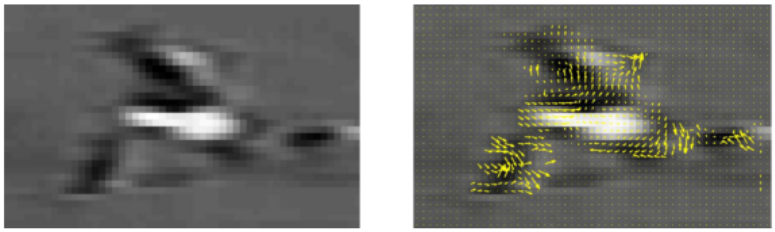

Fig. 2.3: Motion description by the optical flow algorithm - original image (left) and the computed optical flow (right) (source: [73]).

2.2.7 Deep learning

Artificial neural networks (ANN) were inspired by the biological neural networks in human brains. The structure of ANNs consists of multiple connected layers of artificial neurons. Early implementations of ANNs employed only a few layers with several neurons and were efficient in solving relatively simple problems. More difficult problems required deeper and larger networks which proved challenging to learn. Hinton et al. [112] presented an algorithm for fast learning of deep neural networks (DNN), launching the era of deep learning. It [253] it was shown, that the so-called vanishing gradient problem [17] can be solved by using well-designed random initialization and therefore it is possible to train DNNs with stochastic gradient descent algorithm with momentum. Deep convolutional neural networks (CNN) proved to be particularly effective for large-scale image classification and were able to outperform other methods on the ImageNet dataset which contains 15 million images belonging to roughly 22 000 categories [141]. Large-scale visual recognition found applications in web search engines [254]. A comparison of shallow and deep methods for image recognition is provided in [35].

A straightforward approach to applying deep CNNs to action recognition is to employ 3D filters in spatio-temporal volumes rather than employing 2D filters in images. In [135], a sequence of the 2D contours of a person is used to extract features using a CNN with 3D Gabor filters. Action classification is then performed with a weighted fuzzy min-max neural network. The raw images are fed to a 3D CNN in [123], and the network extracts the features automatically (see Fig. 2.4). The authors of [12] first extract the spatio-temporal features with a 3D CNN, and then train a recurrent neural network (RNN) [114] to classify action sequences based on the temporal evolution of the features. Unsupervised learning based on long short-term memory (LSTM) [90] networks enables not only video classification, but also prediction of future sequences, by extrapolating the observed motion [248]. Bi-directional LSTMs are used for action recognition in video sequences in [265].

Neural networks can be provided with multiple streams of data and some layers of the network can be processed separately. The authors of [128] propose a two-stream network with a context stream containing the full images and a fovea stream containing the regions of interest. They also analyze different methods of fusion of the streams - early, late, and slow. Another two-stream network is presented in [245], where a spatial stream encodes information extracted from still images and a temporal stream encodes the motion between frames. Fusion of spatial and temporal streams is addressed in [81].

Some works explored the possibilities of creating hybrid methods, including both deep learning and feature-based approaches. In [261], a 3D CNN is used to extract spatio-temporal features which are then employed for classification based on support vector machines (SVM). The authors of [277] compute convolutional feature maps which are then constrained on the basis of the improved dense trajectories [272]. In regard to time domain most CNNs operate on short time windows, although attempts were made to employ full-length sequences [302]. Recently, deep learning methods are also adapted for other data modalities, including depth [305], skeletal [216], infrared [5] and inertial [105] signals. Multimodal approach is presented in [64]. A surveys on deep learning in action recognition can be found in [109, 281, 309].

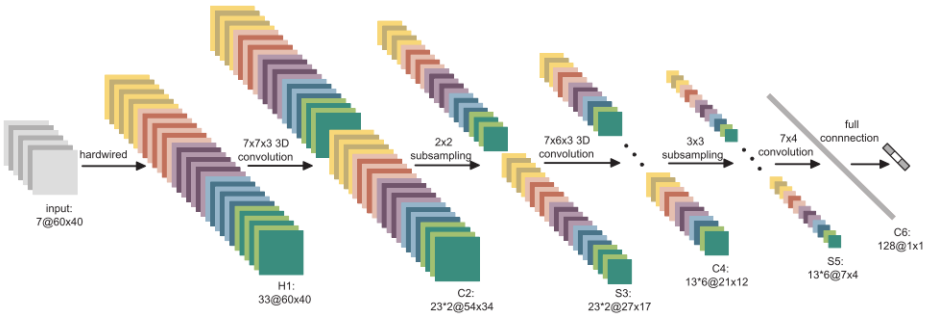

Fig. 2.4: Deep 3D convolutional neural network for human action recognition (source: [123]).

2.2.8 Depth-based methods

Depth sensors are able to measure the distance of the objects in the observed scene. Such data is very useful in motion analysis, as it greatly facilitates the segmentation of objects and provides information about motion along the axis perpendicular to the view plane. Moreover, infrared depth sensors have their own light source, which makes them robust to changing lighting conditions. For these reasons, depth-based methods for motion analysis have been investigated intensely in the recent years as an alternative to solutions based on data from RGB cameras.

There are three main types of depth sensors [45]: stereo cameras, structured light sensors, and time-of-flight (ToF) sensors. Stereo cameras mimic human vision - images from two slightly shifted viewpoints are used for 3D reconstruction. This approach requires only two RGB cameras to compute the depth map, although the calculation cost is significant and varying lighting conditions make this process challenging. Structured light sensors use an infrared light source to project a complex pattern on the scene, which is invisible to human eye, but visible to the sensor. By analyzing the irregularities in the pattern cast on the scene, it is possible to reconstruct the depth maps. Such a solution was employed in the popular Microsoft Kinect [236]. ToF sensors employ active infrared light pulses and measure the time needed for the reflected light to travel back to the sensor. This technique was adapted in the Kinect 2 [236]. Finally, 3D laser scanning devices were used in some studies [20], although they are not very popular due to their inability to provide data in real-time.

There are two main approaches for employing depth information for human motion analysis. One is to operate directly on the depth maps, and the other is to extract a skeleton and make use of the positions of the joints. Depth map approaches include a

variety of different techniques. The authors of [154] create an action graph which encodes actions as a sequence of salient postures, described as a bag of 3D points, extracted from the silhouette. In [307], local spatio-temporal descriptors are extended to 4D by including the time dimension. Action sequences are modelled as 4D shapes in [275], and randomly sampled 4D subvolumes are employed to extract the so-called random occupancy patterns (ROP). Vieira et al. [268] propose space-time occupancy patterns (STOP) - an action representation based on dividing the space-time volume with a 4D grid and identifying the relevant cells in order to encode key poses. The authors of [210] use surface normal orientations of time, depth and spatial coordinates in a 4D space in order to construct a histogram of oriented 4D normals (HON4D) descriptor (see Fig. 2.5). In [156], a rich motion descriptor is created by combining a depth based motion history descriptor called the 3D motion trail model (3DMTM) with the pyramid HOG (PHOG). Recently, convolutional neural networks were employed for depth-based action recognition as well [279, 305].

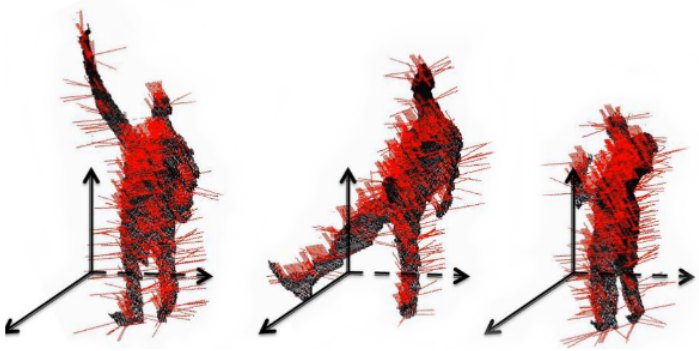

Fig. 2.5: Depth map-based action recognition - surface normals in HON4D descriptor (source: [210]).

Skeleton-based approaches became popular in human motion analysis for several reasons. Johansonn [126] showed in his experiment that actions performed by people can be successfully recognized by analyzing only the motion of the markers corresponding to joints without having to see the entire person. Depth data greatly facilitates the extraction of skeletons and hence the automatic analysis of joint positions. Moreover, processing joint coordinates is much less computationally expensive than processing depth maps or RGB data. Early work focused on the detection of distinctive body parts such as the head, hands, and feet [87]. An efficient method for human pose estimation was presented by Shotton et al. [244]. The authors created a large, realistic, and varied synthetic set of training images, which were used to train a random forest (RF), that allows to efficiently find body joint positions. This algorithm is employed in the Kinect sensor. Authors of [298] compute histograms of 3D joint locations (HOJ3D) as a compact pose representation. In [276], sequences of joint positions in time are transformed to the frequency domain and the fourier temporal pyramid (FTP) descriptor is created. Differences between joint positions in the current, previous, and initial frames are used in [299] together with principal component analysis (PCA) in order to compute the EigenJoints descriptor (see Fig. 2.6). Geometric relations between joints are employed in [274] for learning robust action representations, which are classified with a recurrent neural network. The combination of both depth maps and skeletons is also investigated in several works [269, 276, 308].

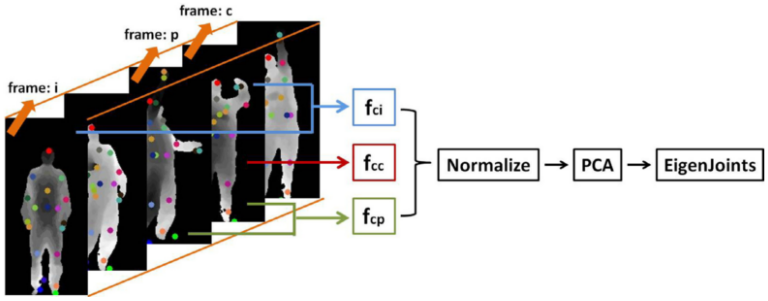

Fig. 2.6: Skeleton-based action recognition: EigenJoints method framework (source: [299]).

2.2.9 IMU-based methods

Inertial measurement units constitute an interesting alternative to visual sensors. Advanced IMUs are able to measure signals from three types of sensors: an accelerometer, a gyroscope, and a magnetometer, in three axes each, resulting in a total of 9 degrees-of-freedom (DOF). IMUs are free of many of the issues related to visual sensing, such as the susceptibility to changing lighting conditions, occlusions or violation of privacy [28]. Moreover, IMUs usually operate at a much higher sampling frequency (250 Hz - 400 Hz for a typical mid-range sensor), which may be important in the analysis of highly dynamic motion. On the other hand, IMUs have to be attached to the examined person, which can be problematic in some scenarios. Other possible issues include the need for calibration, susceptibility to magnetic disturbances, and limited operational time, as wireless sensors relay on batteries. Most importantly, IMUs deliver information about changes in motion, such as acceleration or angular velocity, which makes it difficult to reconstruct the positions of joints due to the accumulation of errors [235]. On the other hand, they can measure orientation with high accuracy, which in some scenarios may be more useful than position.

The first attempts at estimating orientation were based solely on accelerometers [199] or gyroscopes [25]. However, neither of them proved to be sufficient by themselves, therefore a sensor fusion approach was proposed which integrated both the accelerometer and the gyroscope readings [66]. Even more accurate orientation was achieved by combining the accelerometer, gyroscope, and magnetometer by using the Kalman filter [230] and the extended Kalman filter (EKF) [233]. A similar issue to sensor orientation estimation is the analysis of angular kinematics of joints [220], where the goal is to find the relative position of body parts by estimating the angles in the relevant joints. This is usually achieved by employing two IMUs, one on each side of the joint, and measuring the difference between their orientations [207].

Apart from orientation estimation, inertial sensors are also useful for motion analysis [54] and action classification [6]. Since IMUs operate at high frequencies and produce noisy data, a popular approach is to compute statistical features in time windows [10]. Temporal segmentation includes both the detection of activities in continuous signals [202, 247] (see Fig. 2.7) and the division of the identified activity into adjacent or overlapping windows [6, 46]. For each window, the features are usually computed in one or more of the following domains: time, frequency or time-frequency [10]. The time domain features include, among others, mean, root mean square (RMS), standard deviation, variance, and

mean absolute deviation (MAD) [46, 187]. Fast Fourier transform (FFT) is employed to transform the signal into the frequency domain and extract features such as FFT components, energy, and entropy [13]. Wavelet transforms are used to extract wavelet coefficients in the time-frequency domain [206].

With recent development of smartphones and wrist-worn fitness trackers, equipped with inertial sensors, IMU-based motion analysis found applications in physical activity recognition and tracking [195]. In [47] a smartphone-based method, robust to changes in orientation, placement and subject variation is proposed for activity recognition. A comparison of smartphone and smartwatch-based activity recognition is discussed in [288]. Deep learning approaches are applied to inertial signals as well [105].

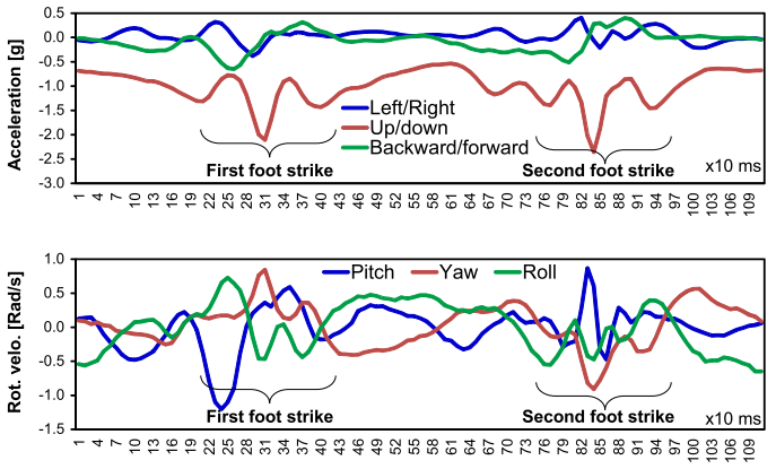

Fig. 2.7: Accelerometer (top) and gyroscope (bottom) signals from IMU attached to hip of walking person (source: [202]).

2.2.10 Multi-sensor systems

Better accuracy of human motion tracking is achievable by employing multiple sensors. Two approaches can be distinguished in this area. The first one is to use multiple sensors of the same type, which is the basis of most of the so-called motion capture (MoCap) systems [243]. The second approach is to use different types of sensors and provide the motion description by fusing the multimodal information.

MoCap systems can be based on either visual [193] or inertial [151] sensors. Visual systems often employ markers which are placed on a person and tracked by multiple cameras from different viewpoints. The markers are usually placed near the joints in order to allow for accurate skeleton reconstruction, and can be either passive or active. Passive markers are made from highly reflective materials and require dedicated external light sources [200], while active markers each have a light source of their own [217]. Markerless MoCap systems usually employ depth sensors, such as the Kinect, which is able to reconstruct the skeleton. However, the accuracy of a single Kinect is often not satisfying, therefore multi-Kinect systems have been proposed [153, 164] (see Fig. 2.8). Full-body motion capture is also possible with an extensive set of inertial sensors [50]. The comparison of motion measurements derived from inertial sensors with a 3D marker-based optical motion capture system is provided in [30].

Since each type of sensor has its own advantages, fusion of multimodal data can give better results. Combining RGB and depth, often referred to as RGB-D, is quite intuitive, as both are visual data. Coordinate mapping between these two modalities is supported by sensors such as the Kinect, although custom methods can also be found in the literature [108]. The analysis of human activities using both RGB and depth is discussed in [140, 152], and a number of RGB-D datasets is publicly available [309]. Multiple RGB-D cameras are employed in [160] and a fusion of RGB, depth, and skeleton data is presented in [241]. Inertial sensors can be used in combination with both RGB and depth cameras. The authors of [247] employ an RGB wearable camera as well as IMUs for temporal segmentation and activity classification. The fusion of depth and inertial data have been recently investigated as well [40, 64, 95]. IMUs can also be used with other types of wearable sensors, such as heart monitors or light sensors [151]. It is worth noting, that data synchronization of the multimodal signals is an important issue [173].

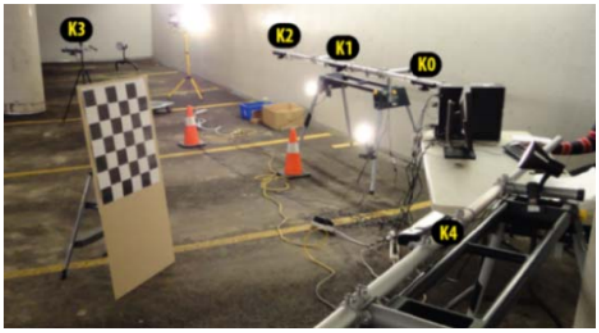

Fig. 2.8: Calibration of multiple Kinect sensors (source: [164]).

2.3 Automatic motion analysis in sports

General and sports motion analysis differ in two crucial aspects. Firstly, in sports, not only action recognition is important, but qualitative analysis of motion as well. While recognition of actions is important in terms of temporal segmentation and the study of tactics, the most interesting information that can be provided for an athlete is how to improve their performance of a technique, particularly in the case of individual sports. This requires systems capable of measuring parameters relevant to the motion. The second aspect is that sports motion analysis is very domain specific. Different types of motions and techniques are used in each discipline, therefore it is difficult to propose general methods for motion analysis which would be suitable for various disciplines. Instead, methods developed for one discipline could be adapted for another by taking into account the specifics of the actions in both of them. In this section, a survey of literature devoted to automatic motion analysis in sports in presented. Then, the discipline selected for this study is discussed and justified. Finally, the objectives of the research are presented.

2.3.1 Overview

The automatic analysis of sports can focus on many aspects. The most general problem is the recognition of sports disciplines, which can be done e.g. by extracting textures from videos along with other cues, such as audio signals [190]. On a more fine-grained level the recognition of specific activities or actions is performed [264]. This usually requires the detection and tracking of athletes as well as the ball in disciplines where

it is used. Once the actions are extracted, performance measures can be devised and qualitative analysis can be provided. This can be used as the feedback for the coaches and the athletes.

The process of the automatic analysis of motion in sports poses a number of challenges, as described in [257]. Inertial sensors need to be attached in a manner which is non-interfering with the athlete's movements. In the case of visual sensors, several issues need to be addressed, such as different viewpoints, the calibration of multiple cameras, a large variation of poses, fast movement, occlusions, changing backgrounds or the similar appearance of players in team sports. There are many aspects on the basis of which sports disciplines can be classified, such as contact vs non-contact sports or the amount of static and dynamic motion components [228]. From the perspective of automatic motion analysis, the most important factor seems to be the division between team and individual sports, as it results in significantly different analysis methods and goals. Therefore, the following survey of the literature in this area is presented with respect to these criteria. Publicly available datasets for sports action analysis are also discussed in a separate subsection due to their important role in the development and comparison of sports analysis methods.

2.3.2 Team sports

There are several important characteristics of team sports which significantly influence the development of automatic analysis methods. Firstly, the presence of multiple players requires simultaneous tracking of multiple objects as well as their identification. Secondly, video broadcasts from team games usually include frequently changing shots from different cameras, and, thus, different perspectives. Finally, team sports usually use a ball, which is difficult to track, as it is small and moves fast. Camera motion, player tracking and ball tracking are therefore the most often addressed issues in this area.

In order to analyze footage from team games, it is necessary to either know camera orientation a priori or detect it automatically. An algorithm for vertical axis detection in sports videos is proposed in [304]. The authors of [44] go further by introducing a method for the automatic planning of camera movement based on the players' current activities. Segmentation of sports broadcasts is addressed in multiple papers and includes fine classification of camera shots [127] as well as the detection of key events for the purpose of summarization [99]. Annotation of sports videos, including both low-level events and high-level semantics is performed in [295]. Crowd analysis can also provide interesting and useful information [242].

Player detection, tracking, and identification is a widely researched topic, particularly in soccer [179]. Commonly addressed issues include occlusions, playfield detection, camera blur and illumination changes [179]. The detection of players is usually performed by background subtraction with some additional processing, such as scene-specific classifier adaptation [213]. Tracking is often based on particle filters combined with techniques such as position estimation [188] or color histograms [136]. Long-term tracking based on generating a tracking hypothesis is addressed in [198]. Player identification is possible by jersey number recognition [88], learning the relative positions of players [88] or combining video data with inertial sensors for motion feature matching [101]. The tracking and identification of players allows for high-level tactics analysis including route patterns and interaction patterns [313].

Ball detection and tracking in team sports is a challenging problem, due to the fast movement of the ball, but also due to occlusions resulting from many players being present in the game. Authors of [158] detect the ball using a combination of color, size and shape cues. Subsequent tracking is performed with Kalman filter-based template matching. In [43] trajectories of ball in the game of volleyball are reconstructed in 3D, based on the input from a single camera (see Fig. 2.9). An alternative approach to ball tracking in soccer and basketball is presented in [282], based on detecting which player is in the possession of the ball. Particle filters were proved to be effective for ball tracking as well [48].

Fig. 2.9: Ball trajectory approximation in volleyball (source: [43]).

2.3.3 Individual sports

Most methods for player detection, tracking, and identification can be successfully applied to many different team sports with little adaptation. The study of individual sports, on the other hand, is much more discipline-specific, as the focus is placed more on qualitative analysis. The performed actions and the motion parameters of interest vary greatly between individual disciplines. In this subsection, relevant work in regard to various individual disciplines is discussed.

The most important goals in individual sports analysis are performance assessment and providing feedback. In [227], high-level analysis of a tennis game is achieved by tracking the players and the ball, as well as the extraction of action sequences to serve as support for the coaches. Deep learning approach to tennis ball tracking is presented in [306]. The timing and body angle consistency of a gymnast during a pommel horse spinning routine is evaluated with the help of the Kinect in [226] (see Fig. 2.10). The Kinect is also employed in [42], for a Yoga self-training system which helps performing the correct poses. The authors of [11] present methods for rapid feedback in three disciplines: rowing, by displaying plots from an accelerometer; table tennis, by the detection and visualization of ball impact positions; biathlon, by measuring the motion of the barrel of the rifle just before and after a shot. Multiple inertial sensors are employed in [4] for technique analysis in basic sports activities such as jogging, sprinting or jumping. Based on a comparison of data from healthy and injured subjects the authors claim that their system can help in the early detection of potential injuries. The tracking and evaluation of a golf swing is a popular research topic and has been addressed by employing: the Kinect with a Gaussian mixture model (GMM) and SVM [310], the Kinect with a particle filter and dynamic programming matching [143], and statistical features from multiple inertial sensors [91]. A full body inertial measurement system was employed in [72] for posture analysis in dressage riding.

Some authors formulate other goals for individual sports analysis, sometimes specific to a particular discipline. The detection and tracking of athletes is important in racing

sports such as canoe/kayak slalom [70] or horse racing [107]. It is worth noting that the movement pattern in racing is significantly different than in team sports, therefore different methods are proposed for both areas. The recognition of general sports activities such as cycling, running, rowing or weightlifting is addressed in [184] by analyzing the similarity of signals from an accelerometer. A summarization of personal sports videos is achieved in [211] by the extraction of highlights in a game of Kendo. In golf, not only the swing motion is analyzed [266], but the kinematics of the human body as well [115, 256]. Oyama Karate techniques are classified by extracting angles from the skeleton data provided by the Kinect [98]. Authors of [124] utilize a convolutional neural network for automatic key frame detection in weightlifting videos, therefore allowing to supervise the athlete's pose during training. An inertial sensor-based, wearable system for supporting ice hockey players training is proposed in [103]. Swimming strokes are counted and the swimming style is identified in [212] on the basis of the accelerometer signal. The authors of [100] propose gesture-controlled interactive video mirrors, which provide simple, instant feedback by displaying the recorded action. Boxing punches are detected and classified from depth images in [129]. A survey on employing inertial sensors for combat sports analysis is given in [296]. The diversity of the discussed methods illustrates how complex and discipline-specific motion analysis in sports can be.

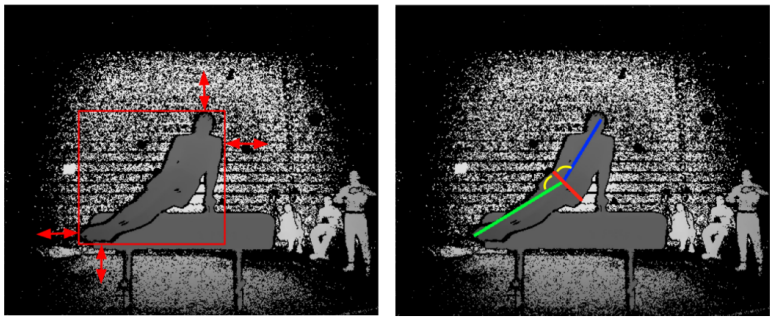

Fig. 2.10: Pommel horse routine analysis (source: [226]).

2.3.4 Datasets

Publicly available datasets are useful for comparing different algorithms on the same input data. Many such datasets have been proposed over the years. Detailed reviews can be found in [34, 309]. The datasets vary in terms of data modality (RGB, depth, inertial), the number and type of actions, difference in conditions (e.g. occlusions), viewpoints, and the number of people in the recordings. There is a number of datasets targeted at specific applications, such as pedestrian detection [77], fall detection [145], gesture recognition [185] or interaction recognition [303]. However, in this section we will consider only the datasets which are related to sports actions either directly, by being designed for sports, or indirectly, by including sports-related actions.

General action recognition is the most popular subject in this area of research, therefore datasets with general actions are the most common. A selection of such datasets, published within the last 15 years, is presented in Table 2.1. While more datasets for general action recognition are available, these were chosen for discussion as they: include sports-like actions, are often used in research (according to the number of references in the literature), and are representative (regarding selection of modalities and actions). The first generation of datasets (KTH [238], Weizmann [94], UIUC [262]) included color data modality only. The selection of actions focused mainly on basic movements such as

Table 2.1: Selected action recognition datasets. #A - number of actions, #S - number of subjects, #E - number of examples (total).

Name	**Modalities #A/#S/#E**	**Actions**
KTH (2004) [238]	Color 6 / 25 / 2391	walking, jogging, running, boxing, hand waving, hand clapping
Weizmann (2005) [94]	Color 10 / 9 / 90	run, walk, skip, jump in place, jumping jack, jump forward on two legs, wave one hand, wave two hands, gallop sideways, bend
UIUC Action (2008) [262]	Color 14 / - / 532	walking, running, jumping, waving, jumping jacks, clapping, jump from sit-up, raise one hand, stretching out, turning, sitting to standing, crawling, pushing up, standing to sitting
MSR-Action3D (2010) [154]	Depth, Skeleton 20 / 10 / 567	high arm wave, horizontal arm wave, hammer, hand catch, forward punch, high throw, draw x, draw tick, draw circle, hand clap, two hand wave, side-boxing, bend, forward kick, side kick, jogging, tennis swing, tennis serve, golf swing, pickup and throw
UTKinect (2011) [298]	Color, Depth, Skeleton 10 / 10 / 200	walk, sit down, stand up, pick up, carry, throw, push, pull, wave and clap hands
G3D (2012) [21]	Color, Depth, Skeleton 20 / 10 / 659	punch right, punch left, kick right, kick left, defend, golf swing, tennis swing forehand, tennis swing backhand, tennis serve, throw bowling ball, aim and fire gun, walk, run, jump, climb, crouch, steer a car, wave, flap and clap
UCF-iPhone (2012) [189]	Inertial 9 / 9 / 383	biking, climbing stairs, descending stairs, gym biking, jump roping, running, standing, treadmill walking, walking
UCFKinect (2013) [75]	Skeleton 16 / 16 / 1280	balance, climb up, climb ladder, duck, hop, vault, leap, run, kick, punch, twist left, twist right, step forward, step back, step left, step right
UTD-MHAD (2015) [38]	Color, Depth, Skeleton, Inertial 21 / 7 / 861	swipe to the left, swipe to the right, hand wave, two hand front clap, throw, cross arms in the chest, basketball shoot, draw x, draw circle (clockwise), draw circle (counter clockwise), draw triangle, bowling, front boxing, baseball swing, tennis forehand swing, arm curl (two arms), tennis serve, two hand push, knock on door, catch an object, pick up and throw, jogging in place, walking in place, sit to stand, stand to sit, forward lunge, squat

walking, running, jumping or waving hands. The release of the Kinect sensor inspired research on depth-based computer vision methods, including the development of action recognition datasets which include depth and skeleton data. Due to the use of depth data, distinction between actions such as throw, push and pull (UTKinect [298]) or step forward and step back (UCFKinect [75]) became feasible. A few sports-related actions are included in some datasets, for instance tennis swing, tennis serve, golf swing, bowling ball throw (MSR-Action3D [154], G3D [21]). The inertial signals were recorded either exclusively (UCF-iPhone [189]) or alongside color, depth, and skeleton data (UTD-MHAD [38]). The UTD-MHAD dataset includes a number of sports-related actions not present in other datasets, for instance basketball shoot or baseball swing.

There is a limited number of datasets focused particularly on sports. These are listed in Table 2.2. Most of the datasets are designed for recognition of high-level sports activities rather than specific actions and are based on TV broadcasts (UCF Sports [229], Olympic sports [203]), YouTube videos (Sports-1M [128]) or smartphone videos (SVW [234]). One dataset (WorkoutSU-10 [201]) includes actual actions - 10 different types of exercises, such as hip stretch or squats. It is worth noting that the creation of sports-related datasets, particularly discipline-specific ones, is a separate challenge in the development of sports analysis methods and has not been well addressed so far.

Table 2.2: Sport action and activity recognition datasets. #A - number of actions, #S - number of subjects, #E - number of examples (total).

Name	**Modalities #A/#S/#E**	**Actions**
UCF Sports (2008) [229]	Color (TV broadcasts) 10 / - / 150	diving, golf swing, kicking, lifting, riding horse, running, skateboarding, swing-bench, swing-side, walking
Olympic sports (2010) [203]	Color (TV broadcasts) 16 / - / 50	high jump, long jump, triple jump, pole vault, discus throw, hammer throw, javelin throw, shot put, basketball layup, bowling, tennis serve, platform (diving), springboard (diving), snatch (weightlifting), clean and jerk (weightlifting), vault (gymnastics)
WorkoutSu-10 (2013) [201]	Color, Depth, Skeleton 10 / 15 / 1500	hip flexion, trunk rotation, lateral stepping, thoracic rotation, hip adductor stretch, hip stretch, curl-to-press, free standing squats, horizontal punch, oblique stretch
Sports-1M (2014) [128]	Color (YouTube) 487 / - / 1M	automatically annotated via YouTube Topic API
SVW (2015) [234]	Color (Smartphones) 30 / - / 4200	archery, baseball, basketball, BMX, bowling, boxing, cheerleading, discus-throw, diving, football, golf, gymnastics, hammer-throw, high jump, hockey, hurdling, javelin, long jump, pole vault, rowing, running, shot put, figure skating, skiing, soccer, swimming, tennis, volleyball, weightlifting, wrestling

2.3.5 Fencing as a discipline of interest

As discussed in the previous subsections, substantial work has been done in the area of automatic motion analysis in sports. Nevertheless, there are still some interesting and unresolved challenges. First of all, even seemingly small variations in the performance of an action can be of great significance for the final performance of an athlete. This is rarely addressed, as most classification methods deal with notably different actions. Secondly, temporal segmentation is indispensable for providing automated feedback in individual sports, while most works consider either pre-segmented actions or cyclic motion which is relatively easy to analyze. Qualitative analysis of the detected actions is important as well, although rarely performed. Moreover, there is a lack of systems which would be able to learn the proper performance of actions and to evaluate trainings based on such pre-learned models. There also seems to be little progress in means of providing feedback to the athletes. Despite the rapid development of multimedia technologies in the recent years, video recordings and text-based statistics and parameters remain the most commonly used manners of providing feedback.

Real-life scenarios are needed in order to address these challenges. In this work, the main requirements for choosing the sports discipline of interest were that it would include non-cyclic motion as well as advanced techniques and body motion patterns, so that a sophisticated performance analysis and feedback system would be justified. Fencing meets these requirements perfectly, as it is highly technical and the sequence of performed actions is unknown a priori, resulting from ad-hoc decisions of the fencer. A single discipline was chosen, rather than multiple, for two reasons. Firstly, as stated before, motion analysis in sports is often discipline-specific, and, rather than devising general methods, it is more beneficial to devise dedicated methods which can be later adapted to other disciplines. Secondly, data acquisition and feedback from coaches are of great importance, therefore it is practical to form a continuous cooperation with a single sports institution rather than trying to collect data from multiple different institutions. It is also worth noting that fencing is particularly well-developed in Poland, as one of the most important forefathers of modern fencing is Zbigniew Czajkowski, a Polish professor. The fencing knowledge which was necessary for this research was based on his publications [56–59] as well as consultations with coaches trained directly by him.

Fencing includes two important elements which can be practiced and analyzed independently, namely footwork (leg actions) and bladework (weapon control). During footwork practice, a fencer learns proper stance and movement. Fencers stand in a sideways position, called the fencing stance, with their weapon arm directed towards the opponent (see Fig. 2.11 left). Forward and backward steps are used for movement. The basic footwork action during an attack is the lunge (see Fig. 2.11 right). There are different kinds of the lunge, as discussed in Chapter 3, which deals with footwork action classification. The detection and analysis of the lunge action is addressed in Chapter 4. There are also other, more sophisticated footwork actions, such as dodging, although, in this work, only the basic footwork actions are addressed - forward and backward steps, as well as different kinds of the lunge.

Fig. 2.11: Fencing stance (left) and lunge (right).

Bladework in fencing is very technical and requires high precision of movement. Two general types of weapons are used: cutting and thrusting. Regarding the first type, both the edge and the tip of the blade can be used to score points, by cuts or thrusts respectively. In sports fencing the only cutting weapon is the sabre (see Fig. 2.12 bottom). In the case of thrusting weapons, only the tip of the weapon can score points, by way of thrusts. The foil and the épée (see Fig. 2.12 top and middle) are the two thrusting weapons used in sports fencing. Épée is the French term for small sword, and is typically associated with the modern version of this weapon, which is characterized by a large bell-guard (hand protection). Some variations of fencing, such as modern classical fencing, employ small swords with small bell-guards in order to provide more historical compliance (see Fig. 2.13). In this work, thrusting weapons are considered, as they are more technical than cutting weapons, and, therefore, require more accurate analysis. Bladework is analyzed in Chapter 5.

Fig. 2.12: Sports fencing weapons: foil (top), épée (middle), sabre (bottom) (source: [215].)

Fig. 2.13: Small sword with historical handle, used in modern classical fencing (source: [291]).

2.3.6 Automatic fencing analysis - related work

Technology-aided analysis of fencing is addressed in the literature in a limited range. The lunge action is analyzed in several papers. The lunge parameters of elite and novice fencers are compared in [92] by way of kinematic analysis of the lunge using stereophotogrammetry. The authors report a considerably larger knee extension during the middle phase of the motion in the case of elite fencers. The structure of the lunge and its performance with regard to the type and complexity of a technical task is investigated in [26] by employing electromyography (EMG) and high-speed cameras. The authors find that complex actions affect the timing of motor patterns. In [194], the biomechanics of the lunge are analyzed by measuring range, speed, and acceleration based on video capture. Correlation between the ability to maintain lumbosacral neutrality during the lunge and performance in terms of speed and acceleration is reported. Body dynamics during the lunge and the fleche actions are compared in [22] using both a motion capture system and a force plate. Results indicate that the fleche action velocity is higher.

In regard to bladework, various aspects of weapon actions are analyzed. The authors of [182] perform weapon action classification by using kinematic data acquired by a motion capture system. The presented methods are able to recognize a number of parry and attack actions with high efficiency. Kinematic determinants of weapon velocity during the fencing lunge are identified in [27]. EMG is employed in [159] to study the influence of different weapon handles. The results show that handle type affects the distribution of muscle strength in the wrist. Authors of [31] distinguish good and poor weapon action executions by applying a neural network to inertial signals segmented with dynamic time warping (DTW).

Apart from lunge and weapon movement analysis, there are also several studies devoted to reaction time of fencers. The response time of novice and elite fencers is compared in [294] with the help of EMG. The findings confirm faster muscle action in experienced fencers. The effects of fencing expertise and physical fitness on action inhibition are studied in [32]. Based on their results, the authors claim that cognitive control benefits from a combination of physical and mental training. Faster stimulus discrimination in top-level fencers is reported in [68] based on a comparison with a control group. Research conducted in [110] shows more efficient dynamic balance control in fencers than static sport athletes and non-athletes.

While many aspects of fencing have been studied in the literature, several challenges remain unaddressed, namely: recognition of similar actions, temporal segmentation and qualitative analysis of detected actions, learning proper action movement, and providing useful feedback in real-time. The motivation behind this work is to investigate these problems and propose relevant solutions.

2.3.7 Research objectives

In the previous subsections new important challenges in automatic sport analysis were identified. Fencing was chosen as a discipline in which these challenges are relevant and can be addressed by developing proper methods. Objectives of this work are divided into three main parts, related to these challenges:

Recognition of action dynamics in fencing footwork

The first part addresses the issue of recognizing similar actions. As stated before, most works regarding action recognition consider actions which differ significantly. Therefore, the proposed methods usually allow for significant variations in the performance of an action, which can be the result of anatomical differences between persons as well as their personal manner of carrying out a given action. In sports, on the other hand, athletes need to perform actions very precisely, as even slight variations in movement can make the difference between two different techniques. In fencing, four types of the lunge action can be distinguished [56, 59], all of which are very similar. Their trajectories are virtually the same, and the defining factor is the dynamics of the movement. By accelerating or decelerating certain parts of the movement, the fencer effectively performs different types of the lunge. Recognizing actions only by dynamics is a challenging task, which requires a novel approach. In Chapter 3, methods based on depth maps and skeleton data provided by the Kinect as well as inertial data from an IMU are proposed for this task. Novel motion descriptors are proposed to extract relevant information from multiple modalities. An efficient feature selection method is introduced as well. The classification is done separately for each feature set at first, and then multi-modal decision-level fusion is employed to improve recognition accuracy.

Real-time detection and analysis of actions in fencing footwork

The second part is dedicated to the temporal segmentation and qualitative analysis of actions. Fencing footwork practice is a continuous movement, in which fencers constantly move forward and backward by using fencing steps, keeping the proper distance from the moving coach, and perform certain actions, such as the lunge, when given the signal. Alternatively, they can practice the footwork on their own, with random forward and backward movement, and performing actions by their own initiative. In order to analyze specific actions, such as the lunge, temporal segmentation of the movement is necessary. Detection of the lunge action is addressed in Chapter 4 by applying both an inertial sensor and the Kinect. A novel signal filtering algorithm is introduced in order to effectively identify action segments, in spite of data disturbances. Methods for qualitative analysis of the lunge action are proposed as well.

Immersive feedback for bladework practice in fencing using augmented reality

The third part focuses on tracking fast movement, learning motion patterns, and providing real-time, immersive feedback for fencing bladework practice. Weapon techniques in fencing, particularly in the case of thrusting weapons, need to be performed very precisely in order to be effective. The fast motion of the blade is difficult to capture and therefore bladework analysis constitutes an interesting and challenging research problem. In Chapter 5, methods for tracking blade movement as well as learning models of proper trajectories for weapon actions are proposed. Evaluation of bladework practice is addressed as well, by comparison with the learned models. Finally, a novel approach to provide immersive feedback is proposed, by using augmented reality semi-transparent glasses which create a mixed view of the virtual and the real world, by overlaying computer-generated objects on the perceived environment. A calibration procedure is devised, which allows to match coordinates between the virtual and the real world.

2.3.8 Summary

Analysis of motion in sports is an important tool for an athlete's training. Applying modern technology to this task allows for new opportunities. Many methods for the description of motion have been proposed over the years, mostly for the task of classification of general actions. In sports, more specific motion analysis is needed, which requires specialized methods. Although many aspects of sports motion analysis have been addressed in the literature, some important challenges still remain. In this work, novel approaches are proposed for the recognition of similar actions, the temporal segmentation and qualitative analysis of actions, as well as learning models of motion patterns and providing feedback by employing augmented reality.

Chapter 3

Recognition of action dynamics in fencing footwork

This chapter discusses the problem of distinguishing between similar actions, which in the context of fencing footwork is an important issue in sports motion analysis. Feature extraction methods relevant for action dynamics analysis are proposed based on signals acquired with the use of an accelerometer and the Kinect. An efficient feature selection algorithm is introduced as well. Fusion of multiple feature sets extracted from multiple modalities is employed. Evaluation is performed on a dedicated dataset of fencing footwork actions, as well as on UTD-MHAD - a publicly available multimodal dataset for action recognition. Experimental results demonstrate that the proposed methods outperform state-of-the-art algorithms on both datasets.

Action recognition is one of the most popular research topics in the area of computer vision [223, 312]. In a typical scenario, a selection of a number of actions is recorded with multiple subjects and optionally with some additional variations, for instance different view angles [34]. Then, various methods are used to extract informative features, which are employed to train classifiers for the recognition of previously unseen cases [287]. The selection of actions for recognition is one of the crucial aspects of such a workflow. Many datasets were created for this purpose and are available publicly, as presented in Section 2.3.4. In most cases, however, the choice of actions to be included in a dataset is rather arbitrary and not justified by any specific scenario. Moreover, the actions are considerably different. In particular, it is relatively easy for a human to recognize any of these actions just by analyzing their trajectories.

This is not always the case for actions in sports. Figure 3.1 presents trajectories of 2 types of step actions as well as 4 types of lunge actions used in fencing (described in detail in Section 3.1.1). As can be seen, the trajectories of lunge actions are very similar and cannot be efficiently distinguished without additional information. Therefore, in this chapter, the analysis of action dynamics is discussed by taking into consideration the temporal structure of the actions. Novel methods for action recognition, specific for this task, are proposed and verified experimentally. Two types of sensors are employed. The first is an IMU, from which a 3 axis acceleration signal is used. The second one is the Kinect, which provides depth maps as well as skeleton estimation in the form of joint positions. Different modalities are used independently to extract informative features.

Finally, fusion of multiple feature sets is applied in order to achieve higher recognition accuracy. The proposed methods were described in three papers [174, 176, 177].

This chapter is organized as follows. Section 3.1 discusses the details of the fencing footwork actions chosen for the classification task. It also presents the structure and acquisition protocol of the fencing footwork dataset recorded specifically for this research. In Section 3.2, the details of the proposed feature extraction, selection, and fusion methods are described. Experimental results are presented and discussed in Section 3.3. The chapter ends with a short summary in Section 3.4.

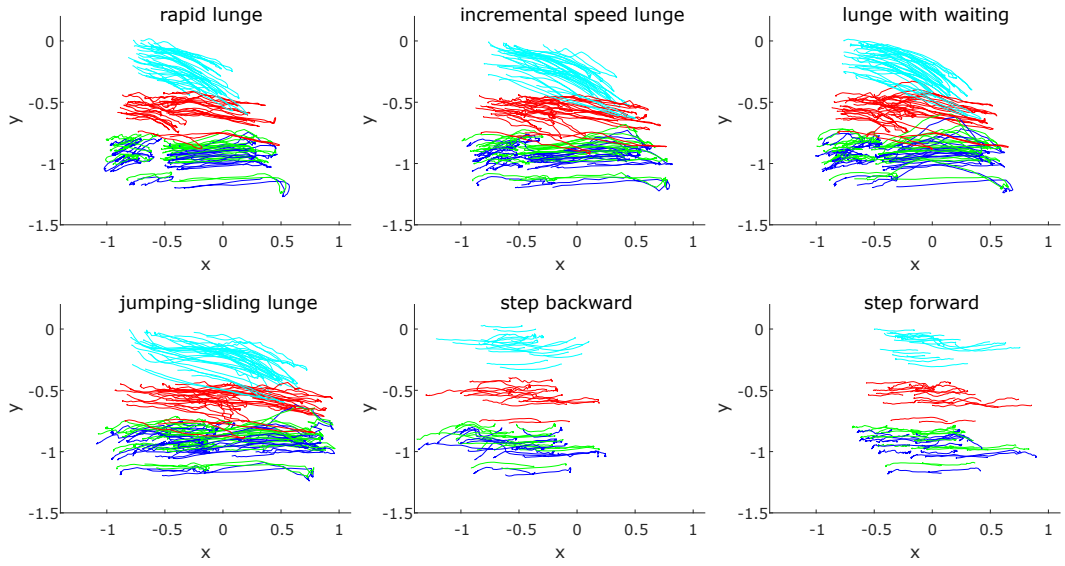

Fig. 3.1: Trajectories of 6 fencing footwork actions for 8 lower body joints: hips (light blue), knees (red), ankles (green), feet(dark blue). Each presented joint trajectory is generated as a mean from multiple repetitions of a given action performed by a single person. Trajectories for a total of 10 persons are presented. Coordinates are given in the Kinect camera coordinate space.

3.1 Fencing footwork dataset

In order to develop automatic sports action analysis methods, a proper dataset is required. Such data are useful not only for developing and testing algorithms, but also for the purposes of comparison with other methods. In Section 2.3.4, an overview of publicly available sports-related datasets is presented. Since there are no publicly available datasets specific to fencing, a dedicated dataset, called the fencing footwork dataset (FFD), was recorded for the purpose of action dynamics analysis in this work. It includes 6 basic fencing footwork actions, which are explained in Section 3.1.1. The acquisition protocol is described in detail in Section 3.1.2. The dataset is publicly available [166].

3.1.1 Fencing footwork

The fencing stance is a sideways position, with the armed hand directed towards the opponent and the other hand kept behind (see Fig. 3.2 left column). The sideways pose is maintained during movement, therefore the front and back legs can be distinguished at all times. The fencer moves forwards and backwards by performing steps. A step forward is done by moving the front foot and then the back foot, thus returning to the

fencing stance (see Fig. 3.2 middle row). A step backward is analogous, but it starts with the back foot and ends with the front foot (see Fig. 3.2 bottom row). During an attack, the fencer performs a fencing lunge (see Fig. 3.2 top row), which is used for rapidly shortening the distance to the opponent. During a lunge, the front foot is slightly lifted, then the front leg is straightened and the back leg energetically pushes the body forward. The full power of the motion is obtained from fully extending the back leg. Returning from the lunge position is performed by bending the knee of the back leg and bringing the front leg back, into the fencing stance. According to fencing coaches and [56], there are four types of lunges:

- Rapid - performed as quickly as possible, usually in a relatively short distance, useful for quick weapon actions
- With increasing speed - starting slowly and finishing quickly, with the speed gradually increasing during the action, useful for feint attacks
- With waiting - includes a short pause in the first phase, during which the fencer waits for the opponent's reaction in order to counter it with their own action
- Jumping-sliding - used in long distances, this lunge resembles a jump, as the fencer strongly pushes with the back leg, which then slides on the floor during the forward motion

The four types of lunges together with the forward and backward steps constitute the basic footwork actions considered in this work. Other footwork actions, such as dodging down or sideways, are not included, as they are significantly different and therefore not relevant for the problem of distinguishing between similar motion patterns. Figure 3.2 presents the key poses of stepping forward, stepping backward, and the lunge actions. The differences between various lunge actions cannot be illustrated simply by key poses, therefore only a single lunge action is depicted.

Fig. 3.2: Key poses of basic footwork actions: lunge (top), step forward (middle) and step backward (bottom). Although there are four types of lunge actions, their key poses are the same, therefore only a single lunge action is depicted.

3.1.2 Proposed dataset

The dataset acquired for this work includes six basic fencing footwork actions, as described in Section 3.1.1:

- rapid lunge (R)
- lunge with increasing speed (IS)
- lunge with waiting (WW)
- jumping-sliding lunge (JS)
- step forward (SF)
- step backward (SB)

In order to verify the applicability of different data modalities for the purpose of fencing footwork analysis, two sensors were employed for the recording, namely the Kinect depth sensor and the x-IMU inertial sensor [255]. The Kinect was placed approximately 3 meters from the person, and recorded the RGB, depth, and skeleton data from the side, at 30 frames per second (see Fig. 3.3). RGB data (640 x 480) were stored in compressed video files. Depth data (16 bit images, at a resolution of 640 x 480) were saved in two 8 bit channels of uncompressed videos. These data include the depth map saved on first 13 bits as well as automatically extracted silhouettes of the person, saved in the last 3 bits. Skeleton data for 20 tracked joints were saved in the MATLAB .mat file format. The IMU sensor was attached to the person's knee and recorded data from the built-in accelerometer, magnetometer, and gyroscope, with a sampling frequency of 256 Hz. Data from the IMU were saved in the .mat file format.

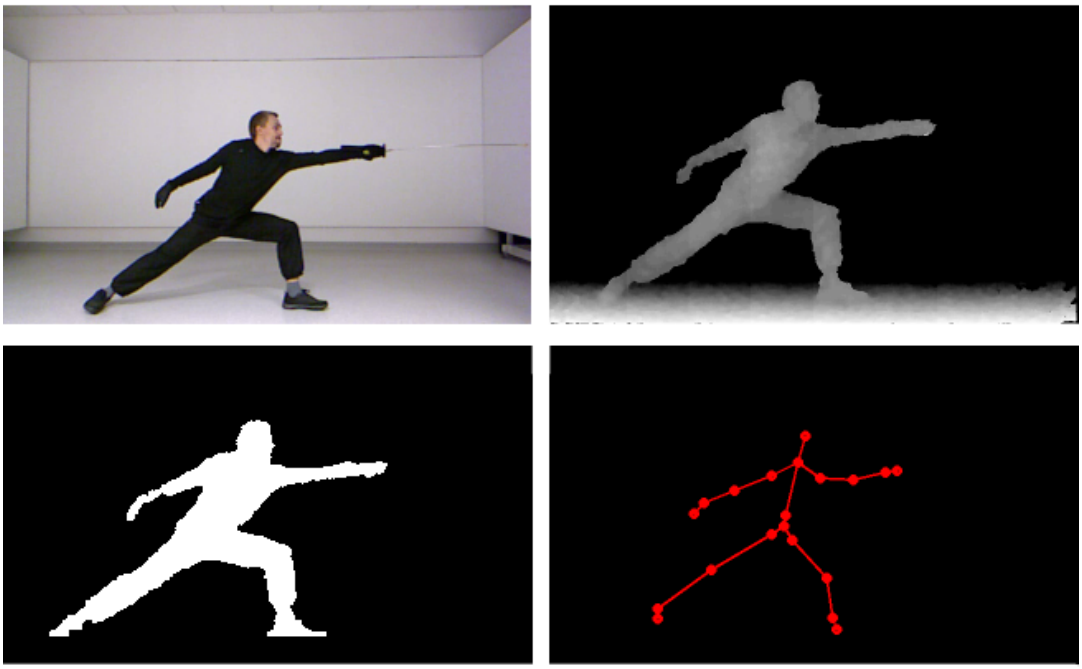

Fig. 3.3: Data provided by the Kinect sensor: RGB (top left), depth (top right), silhouette (bottom left), skeleton (bottom right).

The dataset was recorded thanks to the courtesy of Aramis Fencing School [9], one of the biggest fencing institutions in Poland. Recordings were made of 10 fencers, at levels ranging from intermediate to advanced, both male and female. Every action was repeated by each person 10 or 11 times. Each repetition was recorded separately, in order to avoid the need for temporal segmentation, as the focus was placed on distinguishing fencing actions by their dynamics. The recorded dataset is publicly available: [166]. The RGB data were used to verify recorded actions, but are excluded from the dataset, as for several reasons they are not well suited for sports analysis. Firstly, the lighting conditions in training rooms are often poor, resulting in blurred images, which makes RGB-based analysis of dynamic movement very dependent on these conditions. Secondly, during training, many persons are usually present in the background, making it very difficult for RGB-based algorithms to extract motion of the person of interest. Finally, RGB videos allow to identify persons, which raises privacy issues, an important aspect for many people. The depth and inertial data are free of all these problems.

3.2 Methods

For the purpose of distinguishing similar fencing footwork actions by their dynamics, multiple data modalities are employed, as obtained by the IMU and the Kinect: the 3 axis accelerometer signal, skeleton data, and depth maps. Novel feature extraction methods are proposed in order to capture relevant information regarding the performed motions, from each of these modalities. Accelerometric features are extracted from the IMU data, joint dynamics and local trace images are computed based on skeleton data, and joint motion history context features are based both on the skeleton data as well as the depth maps. Although the proposed features were designed for the analysis of fencing footwork, where the relevant motion occurs on a single plane xy, their extension to 3D is presented as well, by employing 3 orthogonal planes, xy, xz, yz. On the basis of depth data, computing projections for the 3 orthogonal planes is rather straightforward (see Fig. 3.4). Experiments on the 3D extensions of the features were conducted on the UTD-MHAD dataset [38], which contains multimodal recordings of general actions. Even though the proposed features were designed for dynamics analysis in sports actions, they proved to be effective for general action recognition as well. The parameters selected for each method are given for both employed datasets.

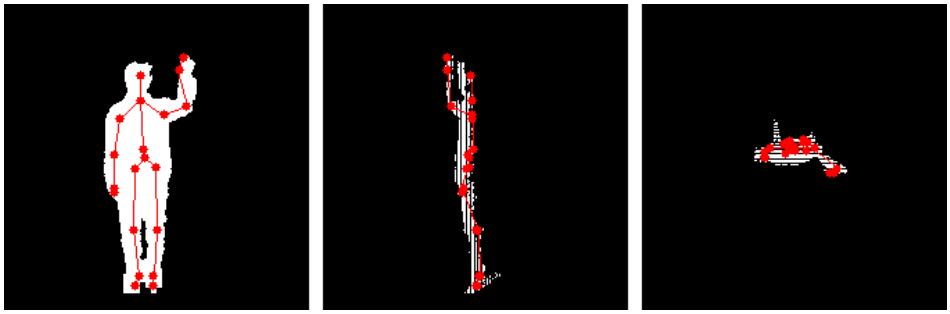

Fig. 3.4: Depth and skeleton data from the Kinect projected onto 3 orthogonal planes.

Since some of the proposed methods generate a large number of features, an efficient feature selection algorithm is introduced, which allows to identify relevant features and obtain better generalization capabilities. Fusion of multiple feature sets, extracted either from one or multiple modalities, has been shown to be beneficial [37]. In this work, both feature-level and decision-level fusion schemes are evaluated, which include not only different modalities, but also multiple feature sets computed from the same modalities.

3.2.1 Accelerometric features

Accelerometric features (Acc) were computed on the raw acceleration data a^{raw}, provided by the IMU sensor, including three components $a_x^{raw}, a_y^{raw}, a_z^{raw}$. In order to remove the gravitational component, the signal from the accelerometer was filtered with the use of a high-pass filter (hpf) with the stopband frequency at 0.4 and the passband frequency at 0.8 normalized frequency units:

$$a^f = hpf(a^{raw}) \tag{3.1}$$

As the information about changes in acceleration and the gravitational component is relevant, the first derivative of the filtered signal a^{df} and the difference between filtered and raw signal a^g were also computed:

$$a^{df} = \frac{d}{dt}a^f \tag{3.2}$$

$$a^g = a^f - a^{raw} \tag{3.3}$$

In order to capture action dynamics, the Acc features were designed to operate in fixed-size time windows. Since the recorded samples differ in length, time normalization was needed. In the FFD dataset, the duration of each recording is close to 2 seconds at a sampling frequency of 256 Hz, therefore each recording was interpolated to a common length of 512 data points. Next, division to time windows was applied. After evaluation of multiple window sizes (32, 64, 128, 256), a size of 128 was selected, with 50% overlap, which resulted in a total of 7 windows per action (see Fig. 3.5). Features were computed in each window separately. For the UTD-MHAD dataset, the duration of the recordings is similar, although the sampling frequency was 50 Hz, therefore the samples were interpolated to 128 data points, and 7 overlapping windows of size 16 were used.

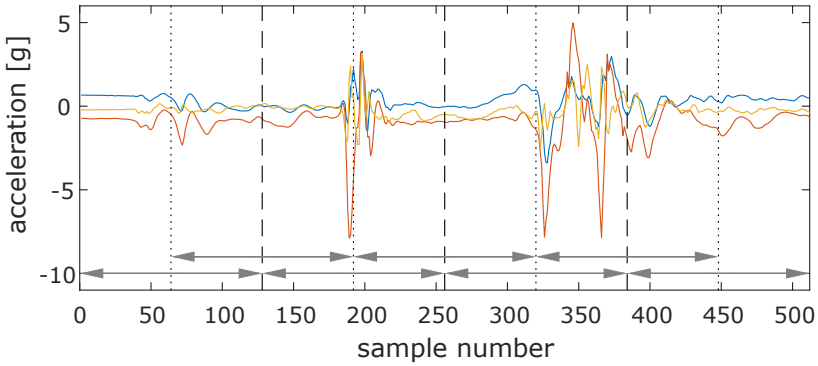

Fig. 3.5: Accelerometer signal (x, y, z) for the incremental speed lunge action. For feature extraction, the signal is divided into 7 overlapping time windows.

A number of accelerometer-based features have been proposed in the literature [151], including time domain [214], frequency domain [13], and wavelet features [102]. In the time domain, statistical features are extracted from time windows, e.g. mean value or the standard deviation. Transformation to the frequency domain is usually obtained by using the fast Fourier transform (FFT) which enables extracting features such as FFT's coefficients. Approximation of the signal is also possible with different wavelet families by finding the proper coefficients, which can be then used as features as well. In this work, each of these domains was studied in order to find the solution best fitted

for action dynamics recognition. The most effective features in each domain were as follows:

- Time domain - using the filtered signal a^f, the difference between the original and filtered signal a^g and the first derivative of the filtered signal a^{df}, statistical measures were computed, namely the mean value and root mean square (RMS) for each axis and magnitude (a total of 24 features per window)
- Frequency domain - using the filtered signal a^f, the difference between the original and filtered signal a^g and the first derivative of the filtered signal a^{df}, the short time Fourier transform (STFT) was computed, and the mean value and RMS of the magnitudes for each axis were taken (a total of 18 features per window)
- Wavelet domain - multilevel wavelet decomposition for the original signal a^{raw} and the filtered signal a^f was computed by using the Daubechies 3 wavelet mother. Next, for each axis, the sums of normalized absolute differences of coefficients for the original signal a^{raw} and the filtered signal a^f were computed, and the sums for levels 3, 4, 5, 6 were used as features (a total of 12 features per window)

From these three domains, the time domain features provided the best accuracy, therefore they were chosen to be used as the Acc features. Given m - mean function, rms - RMS function, a_x, a_y, a_z - x, y, z components of a, a_m - magnitude of a, the Acc features for window w are computed as:

$$\begin{aligned} Acc^w = \{&m(a_x^f), m(a_y^f), m(a_z^f), m(a_m^f)), \\ &rms(a_x^f), rms(a_y^f), rms(a_z^f), rms(a_m^f)), \\ &m(a_x^g), m(a_y^g), m(a_z^g), m(a_m^g)), \\ &rms(a_x^g), rms(a_y^g), rms(a_z^g), rms(a_m^g)), \\ &m(a_x^{df}), m(a_y^{df}), m(a_z^{df}), m(a_m^{df})), \\ &rms(a_x^{df}), rms(a_y^{df}), rms(a_z^{df}), rms(a_m^{df}))\} \end{aligned} \tag{3.4}$$

The final Acc feature vector is the concatenation of features computed in all windows:

$$Acc = \bigcup_{w=1}^{\#windows} Acc^w \tag{3.5}$$

Using 7 temporal windows, the size of the Acc feature vector is 168. It is worth noting that the gyroscope and magnetic data are also sometimes useful in motion tracking [314], particularly for trajectory reconstruction [7]. In this work, both gyroscope and magnetic data were evaluated, but did not improve the recognition accuracy.

3.2.2 Joint dynamics

The Kinect sensor performs real-time tracking of the 3D positions of 20 joints of the human skeleton (see Fig. 3.6). The joint positions are often used for motion analysis by making use of their relative distance in both space and time [276, 299]. However, in this work, in order to focus on action dynamics, velocity and acceleration are considered rather than location, by computing the first and second derivatives of the joint positions respectively. Given $p^j(t)$ - position of $j-th$ joint in time moment t, velocity v^j and acceleration a^j are computed in the following manner:

$$v^j(t) = p^j(t+1) - p^j(t) \tag{3.6}$$

$$a^j(t) = v^j(t+1) - v^j(t) \tag{3.7}$$

The proposed features are called joint dynamics (JD), as they focus on describing the changes in the motion rather than the trajectories. For the purpose of the recognition of actions from the FFD dataset, only the lower body joints are considered, namely the hips, knees, ankles, and feet. The reason is that multiple weapon actions may be performed with identical footwork actions, therefore making upper body motion not relevant in this case. In general, any subset of joints may be selected. Only the xy plane is considered for the FFD dataset, as any motion along z axis is not relevant for the selected actions. In the case of the UTD-MHAD dataset, all joints are employed, and the features for the orthogonal planes xy, xz, yz are computed separately. It is worth noting that including too many joints may lead to redundancy of information and therefore a negative impact on recognition accuracy. This can be handled either by manually selecting the subset or using a feature selection method, as discussed in Section 3.2.5.

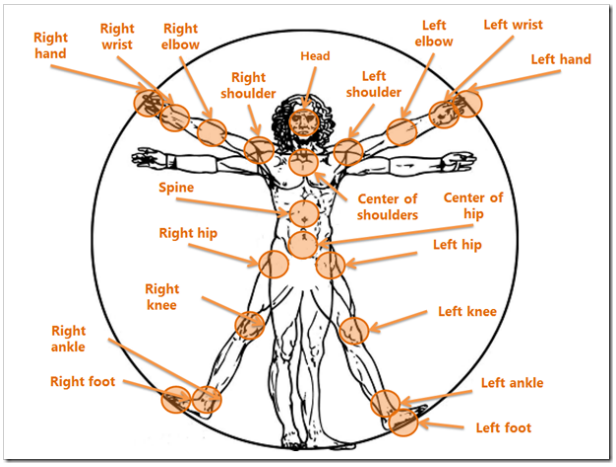

Fig. 3.6: Skeleton joints tracked by the Kinect sensor (source: [192]).

As in the case of Acc features, time windows are employed in order to handle various lengths of recordings and to better model the temporal structure of the actions. However, for the JD, multi-level time division and frequency domain features are employed. The skeleton data for each sample is interpolated in time to 64 data points, as the Kinect acquisition rate is 30 Hz, and the recordings, in both of the considered datasets, are approx. 2 seconds long. Time windows are then employed on multiple levels with a different window size at each level. Based on the experimental study, 3 levels are used, with window sizes 64, 32, 16 and 50% overlap. This division corresponds to 1, 3 and 7 windows respectively, which results in total of 11 windows per action (see Fig. 3.7).

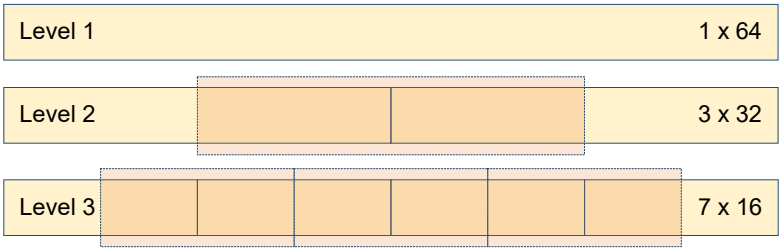

Fig. 3.7: Multi-level time windows for computing JD features.

In each window, a short time fourier transform (STFT) is computed for the velocity and acceleration of each joint along both axes of a given plane. The absolute values of each transform's first 3 coefficients are used as features, resulting in 12 features per joint, per window. Therefore, JD in the xy plane, for joint j and window w are defined as follows:

$$\begin{aligned} JD_{xy}^{j,w} = \{&s^1(v_x^{j,w}), s^2(v_x^{j,w}), s^3(v_x^{j,w}), s^1(v_y^{j,w}), s^2(v_y^{j,w}), s^3(v_y^{j,w}), \\ &s^1(a_x^{j,w}), s^2(a_x^{j,w}), s^3(a_x^{j,w}), s^1(a_y^{j,w}), s^2(a_y^{j,w}), s^3(a_y^{j,w})\} \end{aligned} \tag{3.8}$$

where: $v^{j,w}$ is the velocity of joint j in window w, $a^{j,w}$ is the acceleration of joint j in window w, and s^k is the absolute value of $k-th$ coefficient of STFT transform. The final JD_{xy} feature vector is the concatenation of the $JD_{xy}^{j,w}$ features for all joints in all temporal windows:

$$JD_{xy} = \bigcup_{w=1}^{\#windows} \bigcup_{j=1}^{\#joints} JD_{xy}^{j,w} \tag{3.9}$$

For the FFD dataset, when using a single plane xy, 8 joints, and 11 windows, the total size of feature vector is 1056. For the UTD-MHAD, all 20 joints are employed, and therefore the feature vector size is 2640 per each of the three planes xy, xz, yz.

3.2.3 Local trace images

Instead of using time windows, another approach to motion analysis is to create a single image representing the action. Motion energy images (MEI) and motion history images (MHI) were introduced in [23] and [63]. MEI are simply the sum of the person's silhouette's binary images during the action and represent the region where the movement occurs. In MHI, the intensity of each pixel is computed from the current silhouette and the previous MHI image taken with a decay operator, and therefore it corresponds to the temporal motion at that point. This results in brighter pixels in regions where the motion occurred most recently (see Fig. 3.8 left).

Changes in velocity and acceleration during the motion need to be considered in order to analyze action dynamics. By computing the average intensity of each pixel in superimposed silhouette images rather than using the decay operator, an energy image is created, where brighter pixels represent slower motion (see Fig. 3.8 center). However, employing silhouettes generates noisy images, where relevant information is often lost due to the overlapping motions of different body parts. Therefore, a different approach based on skeleton data is proposed in this work. The positions of tracked joints are modeled by a two-dimensional normal distribution:

$$b(x, \mu, \sigma) = \frac{1}{\sqrt{2\sigma^2\pi}} exp - \frac{(x-\mu)^2}{2\sigma^2} \tag{3.10}$$

where b denotes pixel brightness, given as a function of x (distance to the joint), μ (position of the joint) and σ (variance). By superimposing joints modeled by such distributions, a trace image is constructed (see Fig. 3.8 right) where the spatio-temporal motion patterns of joints are clearly visible. Nevertheless, some relevant information may still be lost due to overlapping motion of different joints. Therefore, local trace images (LTI) are proposed, where each joint's movement is modeled separately. First, a separate, large image is created to represent each joint by superimposing the Gaussians generated for this joint. Then, a minimal square containing all non-zero pixels is selected

and constitutes a single LTI. Such images are then resized to a common size, which was experimentally defined as 16 x 16. In order to construct the final motion descriptor, the images are concatenated (see Fig. 3.9).

LTI can be computed for any given subset of joints and for each of the three orthogonal planes xy, xz, yz. Similarly, as in the case of JD features, 8 lower body joints in the xy plane are employed for the FFD dataset (final feature vector size is 2048) and all joints in 3 planes xy, xz, yz are employed for the UTD-MHAD dataset (final feature vector size is 5120 per plane). It is worth noting, that even though motion pattern images are generated during the computation of the LTI descriptor, the method requires only skeleton data and does not make use of depth maps.

Fig. 3.8: Incremental speed lunge action represented as a single motion image of the lower body: MHI (left), energy image (middle), trace image (right).

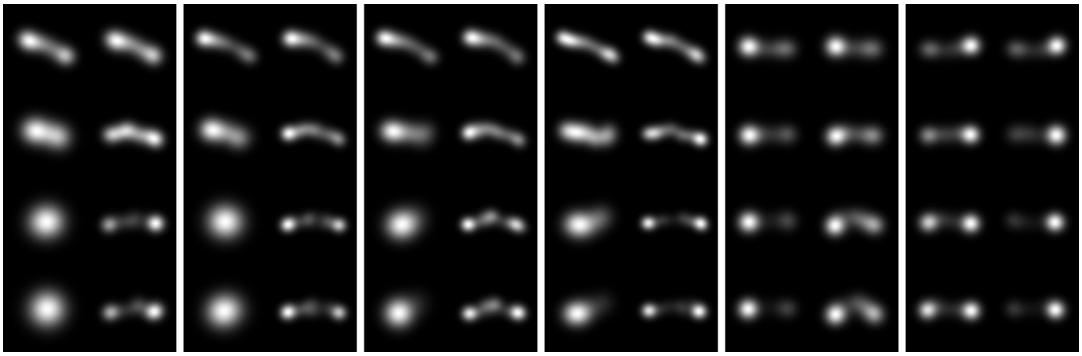

Fig. 3.9: LTI motion descriptor for 8 lower body joints (hips, knees, ankles, feet) and 6 fencing footwork actions, from left to right: rapid lunge (R), incremental speed lunge (IS), lunge with waiting (WW), jumping-sliding lunge (JS), step forward (SF), step backward (SB).

3.2.4 Joint motion history context

The relative motion of body parts is often an important cue in action recognition [259]. Therefore, in this work, a new descriptor is proposed, which is able to capture local motion changes around the selected joints. The motion is detected by making use of differences between silhouette images, which are described with histograms. In order to focus on the relations between local body parts, log-polar histograms are employed. The algorithm was inspired by the shape context descriptor [16], which is commonly used to describe shapes on the basis of object edges. However it differs, as silhouette differences are used rather than edges, and the histograms are computed in the joint positions.

The proposed method employs both depth and skeleton data provided by the Kinect sensor. Depth data contains the extracted silhouettes of the person. The absolute difference in silhouettes between two consecutive frames corresponds to the motion performed in time between the images (see Fig. 3.10 left and center). The joint positions, extracted

from the skeleton data, are used as the center points for the histograms describing the motion (see Fig. 3.10 right). Pixels falling into each bin are counted, and then the histograms are normalized, so that sum of all bins is equal to one.

Given N points $p_1, p_2, \ldots, p_N$ representing the silhouette changes between two consecutive depth maps, the motion context of the $i-th$ joint located in point q_i is defined as a log-polar histogram h_i of the relative points, that indicate the silhouette motion:

$$h_i(l, \varphi) = \#\{p_j \,|\, i \neq j, (log(p_j - q_i), angle(p_j, q_i)) \in bin_{l,\varphi}\} \tag{3.11}$$

where l - log coordinate and φ - polar coordinate in the histogram.

Given N_i - total number of points in histogram h_i, N_b - number of bins in a single histogram, b_k - bins, the normalization of a histogram is performed as follows:

$$\forall k \in 1, ..., N_b, b_k = b_k / N_i \tag{3.12}$$

Therefore each histogram forms a probability distribution:

$$\sum_{k=1}^{N_b} b_k = 1 \tag{3.13}$$

Based on experiments, the size of a single histogram is 12 x 5 for polar and log coordinates respectively. This produces 60 values per joint. Histograms computed for each joint are concatenated, resulting in the joint motion context (JMC) descriptor.

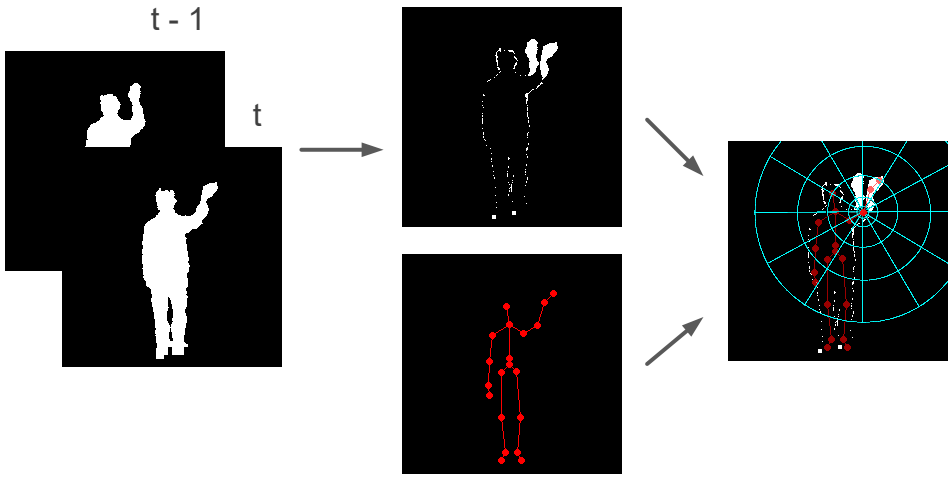

Fig. 3.10: JMC descriptor. Depth data for current and previous frames include silhouettes for both frames (left), which is used to generate silhouette difference (center). Based on the silhouette difference and the skeleton data for current frame, log-polar histograms around selected joints are computed (right).

While the difference between two consecutive frames carries relevant information about the motion, additional information may be extracted by employing earlier frames. In the proposed method, JMC descriptors are computed between the current and several preceding frames separately, and then their weighted sum is calculated, forming the joint motion history context (JMHC) descriptor (see Fig. 3.11). Based on experiments, the number of employed preceding frames is three, and the weight for each frame is: w_{t-3} = 0.25, w_{t-2} = 0.5, w_{t-1} = 0.25. JMHC can be computed for any subset of joints and

in the three orthogonal planes xy, xz, yz. As in the case of JD and LTI features, the motion of 8 lower body joints in the xy plane is considered for the FFD dataset (480 features per frame), and for the UTD-MHAD, all joints in all three planes xy, xz, yz are employed (1200 features per frame in a single plane).

Since the JMHC descriptor is computed per frame, rather than per recording, it requires further processing before the final feature vector is calculated. For this purpose, division to temporal windows is employed, similarly as in the case of Acc and JD features. The vector of JMHC features computed for each frame is interpolated to a length of 64 with single-level time windows of size 16 with 50% overlap, resulting in a total of 7 windows. In each temporal window, statistical measurements are computed from the JMHC features, namely the mean and RMS. Therefore, in each window, the number of features is equal to twice the number of features per single frame. The descriptors for each time window are then concatenated, forming the final feature vector. For the FFD dataset, the final feature vector size is 6720 and for the UTD-MHAD dataset, the final feature vector size is 16800. Due to the large number of features, a selection method is needed, as described in Section 3.2.5.

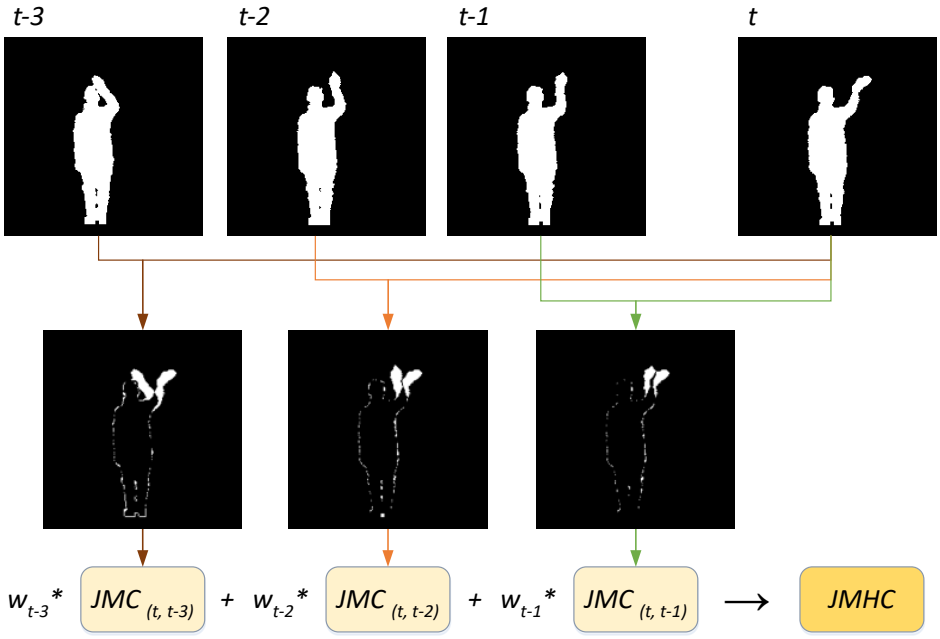

Fig. 3.11: JMHC descriptor is computed as the weighted sum of JMC descriptors calculated between the current and multiple previous frames.

3.2.5 Feature selection

Employing too many features may hinder classification effectiveness [121]. Correlated features provide no extra information, but increase the dimensionality, making it more difficult to train classifiers [33]. Some features may contain more noise than actual information, which can lead to a situation where the classifier makes decisions based partially on random noise data. This occurs particularly, when a large number of features is generated by the descriptor. In order to deal with irrelevant features a selection process may be performed. The additional benefits of reducing the number of features include better generalization and shorter training time [97].

The main approaches for feature selection are wrapper, filter and embedded methods [33, 97]. Wrapper methods evaluate the subsets of features using a given classifier by performing learning and evaluation for each subset. This process is often very time-consuming and prone to overfitting, which occurs when the classifier recognizes training samples very well, but fails to generalize and can't properly recognize test samples. Filter methods, on the other hand, perform feature selection independently of the classifier, usually based on the correlation between the features. In the embedded selection methods, feature selection and classification are combined in order to reduce selection time while maintaining the correspondence to the given classifier. It is worth noting that instead of feature selection, dimensionality reduction can be employed as well, by projecting the features to a lower dimensional space based on their correlation.

Two state-of-the-art feature selection methods, AdaBoost [178, 270] and Lasso [178, 260], as well as a commonly used dimensionality reduction algorithm, the principal component analysis (PCA) [1], were evaluated during experiments concerning the proposed features. Next, a novel filter feature selection algorithm was designed. It employs feature ranking approach [33], although it considers correlation between classes, rather than only between features. It also reduces the redundancy during the selection process, by computing the correlation with the already selected features.

The features are initially ranked in accordance with their inter-class discriminative power determined on the basis of distances between histograms computed for each feature per each class. In each iteration, the correlation between the already selected and the remaining features is updated and taken into consideration when selecting the next feature in order to reduce the redundancy in the final selected feature subset. The details of the algorithm are as follows:

Given:
F - set of features
C - set of classes
P - set of distinctive pairs of classes

$size(P) = \frac{size(C)*(size(C)-1)}{2}$

Distance metric d between two histograms h_1 and h_2, with N_b bins each:

$d(h_1, h_2) = \sum_{i=1}^{N_b} abs(h_1(i) - h_2(i))$

Step 1:
Compute matrix H of normalized histograms per each feature and per each class. Each histogram is a probability distribution of a given feature in a given class.

	1	2	...	$size(C)$
1	$h_{1,1}$	$h_{1,2}$	...	$h_{1,size(C)}$
2	$h_{2,1}$	$h_{2,2}$	...	...
...	...	...	...	...
$size(F)$	$h_{size(F),1}$	$h_{size(F),2}$	...	$h_{size(F),size(C)}$

where:
$h_{i,j}$ = histogram for $i-th$ feature and $j-th$ class
$size(H) = size(F) * size(C)$

Step 2:
Compute matrix A of weights a describing the discriminative power of each feature for each pair of classes:

for i = 1:size(F)
    for k = 1:size(P)
        c_1 = *first class of k-th pair*
        c_2 = *second class of k-th pair*
        $a_{i,k} = d(h_{i,c_1}, h_{i,c_2})$

where:
$a_{i,k}$ = weight a for $i-th$ feature and $k-th$ pair of classes
$size(A) = size(F) * size(P)$

Step 3:
Initialize vector B of weights b describing the correlation between each feature and the already selected features:

for i = 1:size(F)
    b_i = *0*

where:
$size(B) = size(F)$

Step 4:
Iteratively select m features, going over the list of distinctive pairs of classes, by choosing the most discriminative features for each pair, on the basis of the sum of weights A and B. In each step, update weights B by adding the sums of distances between the remaining features and the last selected feature, and normalizing weights B accordingly. Given: F_S - set of already selected features, F_R - set of remaining features, from which the selection is performed, i - iteration, the algorithm is as follows:

$F_S = \emptyset$
$F_R = F$
$i = 0$

while $size(F_S) < m$
    for k = 1:size(P)
        $f_{next} = \underset{f \in F_R}{argmax}(a_{f,k} + b_f)$
        $F_S = F_S \cup f_{next}$
        $F_R = F_R \setminus f_{next}$
        for f = 1:size(F_R)
            $b_f = \frac{i}{i+1} * b_f + \frac{1}{i+1} * \sum_{c=1}^{size(C)} d(h_{f,c}, h_{f_{next},c})$
        $i = i + 1$

3.2.6 Fusion and classification

Two state-of-the-art classifiers were used to verify the usefulness of each feature set separately, namely support vector machine (SVM) [52] and random forest (RF) [113]. SVM calculates the best hyper-plane separating data samples in a multi-dimensional space. Regularization parameter C creates a margin allowing for the misclassification of some samples during the training stage in order to prevent overfitting. For non-linear problems, different kernels may be used with SVM. In this work, the radial basis function (RBF) kernel was employed, which requires the additional parameter *gamma*, corresponding to the range of influence of a single training data point. The values for both parameters were determined with a grid search (see Section 3.3). The RF classifier creates a number of decision trees with random thresholds for feature values in each node. Classification is based on the mode of the classes returned by all trees. The number of trees ($nTrees$) can be set as a parameter of the RF classifier.

Initial experiments with separate feature sets showed that different features are better suited for recognizing different classes. Therefore, feature fusion was employed in order to take advantage of the information provided by all feature sets. In the feature fusion experiments, SVM classifier was used, as it produced better results than the RF classifier. There are two main approaches to fusion - feature-level (early) fusion and decision-level (late) fusion [181]. In feature-level fusion, the features are simply concatenated and provided to the classifier as a single vector. In decision-level fusion, each feature set is given as a separate vector and the classifier decides how to combine the data. In this work, both approaches have been verified. For feature-level fusion, concatenated features were fed to the SVM classifier (see Fig. 3.12). For decision-level fusion, a separate SVM model was trained for each feature set. Weka implementation of SVM was employed [132, 221, 289], which outputs the probabilities for each class rather than only the most probable class. Probabilities from all SVMs were then used as input for an artificial neural network, namely multilayer perceptron (MLP) [231], which is known to be effective for problems with a relatively small number of input data. A number of parameters can be set when training MLP, namely the number of hidden layers, number of neurons in each layer, learning rate, and momentum. The values for the parameters were determined in a grid search (see Section 3.3). The architecture of decision-level fusion is depicted in Fig. 3.13.

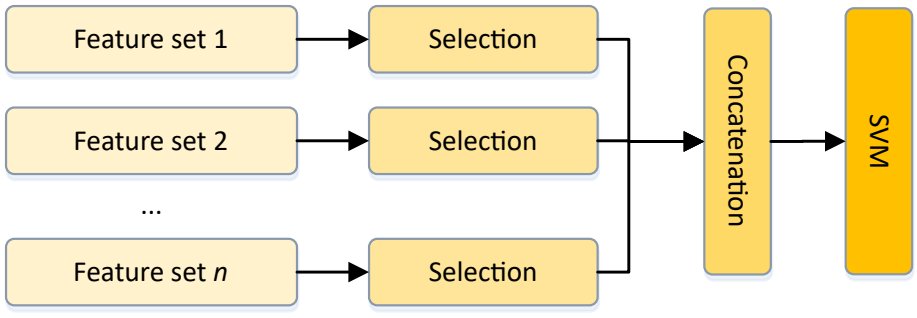

Fig. 3.12: Feature-level fusion scheme - a single SVM is trained on the concatenated feature vector.

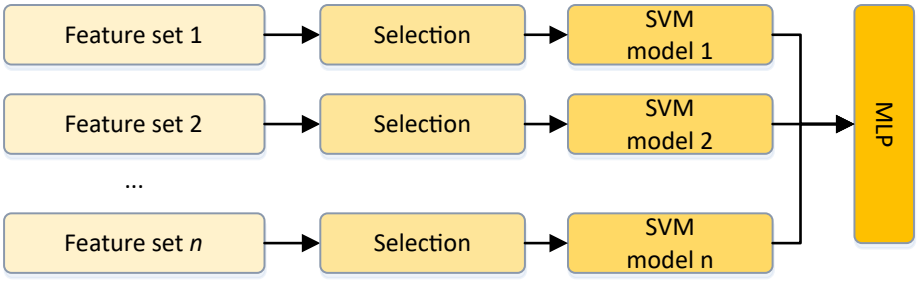

Fig. 3.13: Decision-level fusion scheme - separate SVMs, trained for each feature set, output probabilities for each class, which are then used as the input for MLP.

3.3 Experiments and results

The proposed methods were verified experimentally on two datasets - the proposed FFD dataset and the publicly available UTD-MHAD dataset [38]. Descriptions of both datasets are provided in Section 3.3.1. Results and discussion are provided in Sections 3.3.2 and 3.3.3 for the FFD and UTD-MHAD datasets respectively.

3.3.1 Datasets

The proposed FFD dataset is described in detail in Section 3.1.2. It contains six fencing footwork actions (*step forward (SF), step backward (SB), rapid lunge (R), lunge with increasing speed (IS), lunge with waiting (WW), jumping-sliding lunge (JS)*), recorded by 10 fencers, with 10 to 11 repetitions. The acquired data includes depth and skeleton data recorded with the Kinect, as well as an accelerometric signal, recorded with the x-IMU inertial sensor. The employed evaluation protocol included two scenarios: person-dependent (PD) and person-independent (PI). In the PD scenario, the classifiers were evaluated using five-fold cross-validation for each person separately - for each fencer, in each fold, 80% of data were used for training, and 20% for testing. In the PI scenario, ten-fold leave-one-out cross-validation was employed, where, in each fold, nine subjects were used for training, and one was used for testing. The PD case allows to verify if the actions of a given person were consistent, while the PI case verifies the proposed methods' ability to generalize. Sample frames are illustrated in Fig. 3.14

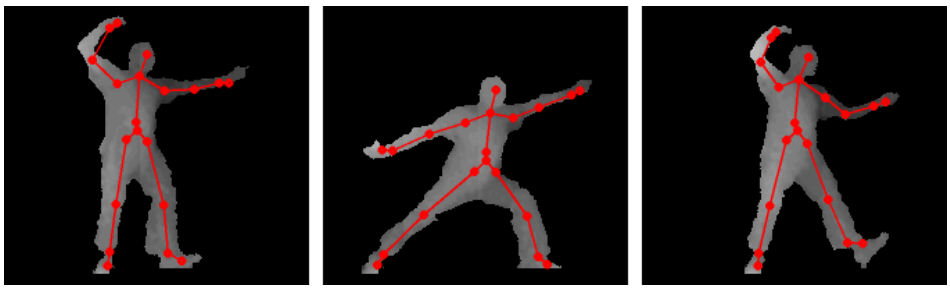

Fig. 3.14: Sample frames from the FFD dataset (depth and skeleton view). Left to right: fencing stance (pose present in all actions), lunge (pose present in all lunge types), step forward.

UTD-MHAD [38] is a publicly available dataset, which includes RGB, depth, skeleton, and inertial data, acquired by the Kinect and a custom low-cost inertial sensor. There

are 27 actions in the dataset, with either one-hand, two-hand, or leg motion: *swipe left, swipe right, wave, clap, throw, arm cross, basketball shoot, draw X, draw circle clockwise, draw circle counter clockwise, draw triangle, bowling, boxing, baseball swing, tennis swing, arm curl, tennis serve, push, knock, catch, pickup throw, jog, walk, sit to stand, stand to sit, lunge, squat.* The actions are performed by 8 subjects with 4 repetitions. The evaluation protocol states to use subjects 1, 3, 5, 7 for training and subjects 2, 4, 6, 8 for testing. Sample actions are presented in Fig. 3.15

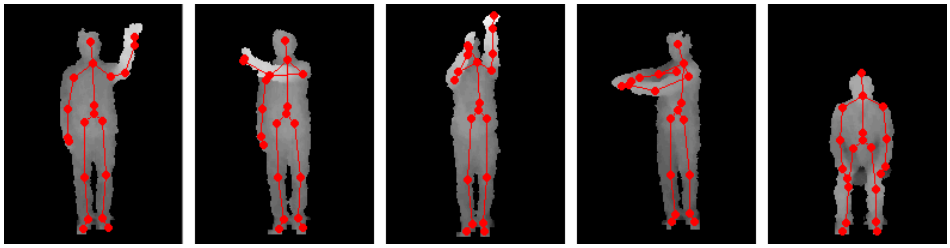

Fig. 3.15: Sample actions from the UTD-MHAD dataset (depth and skeleton view). Left to right: wave, swipe left, basketball shoot, baseball swing, sit to stand.

3.3.2 Results on FFD dataset

The evaluation of the FFD dataset began with the comparison of the SVM and RF classifiers, for both PD and PI scenarios, for all feature sets separately. For the SVM classifier $C = 1$ was used (considered values were 0.1, 0.3, 0.6, 1, 3, 6, 10) and for the RF classifier $nTrees = 150$ was employed (considered values were 10, 30, 50, 70, 100, 150, 200). The results are presented in Table 3.1. In the PD scenario, SVM was better for Acc and JD features, while RF was better for LTI and JMHC features. In the PI scenario, SVM was superior or equal for all feature sets. The PI case is more important (PD is used mostly for the purposes of validation of the dataset), therefore SVM was employed in all subsequent evaluations. It is worth noting that in the case of Acc features another classifier was also used, namely dynamic time warping (DTW) [18]. DTW is commonly used for comparisons of time-series. It resulted in 98.18% and 56.75% accuracy for the PD and PI scenarios respectively, which indicates high efficiency for a single subject, but poor generalization ability. Due to very low accuracy in the PI scenario, DTW was not considered in further work.

Table 3.1: Recognition accuracy (%) for the FFD dataset using SVM and RF classifiers in PD and PI scenarios, for all proposed feature sets.

	SVM		RF	
	PD	PI	PD	PI
Acc	**93.88**	**70.71**	90.58	63.73
JD	**93.55**	**79.82**	92.07	**79.82**
LTI	94.05	**74.62**	**94.21**	72.34
JMHC	86.12	**74.16**	**90.58**	71.43

The next step in the evaluation process consisted of feature selection. Firstly, manual selection was performed, in order to determine the redundancy in the information provided by multiple lower-body joints. For this purpose, experiments were performed with reduced sets of joints, by removing, in turn, features corresponding to hips, knees, ankles, and feet. The results are presented in Table 3.2. Acc features were not considered

in this experiment, as they are based on a single joint. For the JD and LTI features, best results were obtained without using the features corresponding to the ankles. For the JMHC features, the best accuracy in the PI scenario was achieved when using all of the joints. Therefore, in subsequent experiments, JD and LTI features generated without ankles were used, and for JMHC all joints were employed.

Table 3.2: Recognition accuracy (%) for the FFD dataset using the SVM classifier with manual selection of features.

	JD		LTI		JMHC	
Features	PD	PI	PD	PI	PD	PI
All	93.55	79.82	94.05	74.62	86.12	**74.16**
W/o hips	92.23	79.06	92.40	74.42	86.78	68.94
W/o knees	93.06	77.69	92.56	70.71	**86.94**	73.48
W/o ankles	**94.21**	**80.42**	**94.88**	**77.24**	85.45	71.82
W/o feet	93.55	79.36	93.55	74.36	86.45	71.52

Automatic feature selection was evaluated using three different algorithms - AdaBoost [178, 270], Lasso [178, 260] and the proposed method - as explained in Section 3.2.5. The experiments also included dimensionality reduction with PCA [1]. Each feature set was evaluated separately. For each feature set, experiments were conducted for multiple numbers of selected features. Results for the PD and PI scenarios are given in Tables 3.3 and 3.4 respectively. In the PD scenario, the proposed method is superior for all feature sets, except the LTI, for which Lasso is slightly better. In the PI scenario, the proposed method provides significantly better accuracy for the JD and JMHC features. Also, for the Acc features, it is better than other methods, although, in this case, the best results were obtain without any selection. For the LTI features, Lasso selection proved to be better.

Table 3.3: Recognition accuracy (%) for the FFD dataset using the SVM classifier, in the PD scenario, with different feature selection methods used for the proposed feature sets.

		w/o sel.	PCA	AdaBoost	Lasso	Proposed
Acc	#feat	168	40	120	120	120
	acc.	93.88	92.73	93.72	94.55	**94.71**
JD	#feat	792	45	700	600	500
	acc.	93.55	93.72	93.55	94.38	**95.05**
LTI	#feat	1536	100	600	500	800
	acc.	94.05	92.07	95.04	**95.21**	94.88
JMHC	#feat	6720	100	750	650	720
	acc.	86.12	93.55	90.08	93.39	**93.72**

Table 3.4: Recognition accuracy (%) for the FFD dataset, using the SVM classifier, in the PI scenario, with different feature selection methods used for the proposed feature sets.

		w/o sel.	PCA	AdaBoost	Lasso	Proposed
Acc	#feat	168	60	150	160	140
	acc.	**70.71**	68.24	70.06	70.36	70.67
JD	#feat	792	200	400	600	120
	acc.	80.42	78.12	79.18	80.24	**82.52**
LTI	#feat	1536	100	600	500	800
	acc.	77.24	74.16	76.9	**79.03**	77.36
JMHC	#feat	6720	100	750	650	720
	acc.	74.16	75.99	73.1	76.44	**79.03**

The next step of the evaluation considered feature fusion. Two fusion schemes were employed - feature-level fusion and decision-level fusion. In feature-level fusion, the selection was first performed on each feature set, then the selected features from all sets were concatenated in order to form a single feature vector. This vector was then fed to an SVM classifier. Decision-level fusion also began by performing selection for each feature set, but the selected features from each set were evaluated separately by dedicated SVM classifiers. Each SVM classifier returned the probabilities for each class which were concatenated and fed to the MLP classifier, as described in Section 3.2.6. MLP was set to have a single hidden layer, while the other parameters were determined by a grid search: number of neurons (6 to 24 with step 3, selected value 24), learning rate (0.1 to 0.6 with step 0.1, selected value 0.2), and momentum (0.1 to 0.4 with step 0.1, selected value 0.3). The procedure was run for all considered feature selection methods, as well as for the PCA. The results for the PD and PI scenarios are given in Tables 3.5 and 3.6 respectively. The inclusion of Acc features is treated as a separate case, as it requires an additional device, which has considerable impact on the potential applications.

In the PD case (see Table 3.5), feature-level fusion proved to be significantly better, achieving 98.74% accuracy when using the proposed feature selection method. For PCA and Lasso, additional modality (Acc) was beneficial, but it did not influence the accuracy in the case of AdaBoost as well as the proposed method. Overall, the results for the PD case indicate that the actions performed by the fencers were consistent for each person.

In the PI scenario (see Table 3.6), decision-level fusion provided superior recognition accuracy, achieving 86.31% when using the proposed feature selection method. Although additional modality (Acc) was useful when no selection or Lasso selection were employed, for other methods, including the proposed one, it had negative impact on the final result. Most likely, a single accelerometer was not able to provide rich enough data to achieve generalization for this modality - as indicated in Table 3.1 Acc features were the least efficient in the PI scenario. On the other hand, it is worth noting, that final recognition accuracy (86.31%), which employed depth and skeleton modalities, processed by three different feature extraction methods and combined with decision-level fusion, is significantly better than the best accuracy for a single modality - 82.52% provided by JD features after selection (see Table 3.4).

Table 3.5: Recognition accuracy (%) for the FFD dataset, PD scenario, using the fusion of multiple feature sets with respect to different feature selection methods. Feature selections were made for each feature set separately.

	w/o sel.	PCA	AdaBoost	Lasso	Proposed
Feature-level fusion					
LTI + JD + JMHC	90.08	97.3	94.59	97.3	**98.74**
LTI + JD + JMHC + Acc	90.25	98.28	94.59	98.02	**98.74**
Decision-level fusion					
LTI + JD + JMHC	91.89	**93.87**	91.35	92.79	92.79
LTI + JD + JMHC + Acc	92.79	**94.23**	93.15	93.51	93.33

Table 3.6: Recognition accuracy (%) for the FFD dataset, PI scenario, using the fusion of multiple feature sets with respect to different feature selection methods. Feature selections were made for each feature set separately.

	w/o sel.	PCA	AdaBoost	Lasso	Proposed
Feature-level fusion					
LTI + JD + JMHC	79.33	78.52	82.13	82.79	**83.61**
LTI + JD + JMHC + Acc	79.48	78.85	81.31	**82.95**	**82.95**
Decision-level fusion					
LTI + JD + JMHC	83.28	80.16	82.46	83.61	**86.31**
LTI + JD + JMHC + Acc	83.44	79.67	82.13	84.10	**85.66**

For the SVM classifier, it is possible to use different kernels and therefore conduct non-linear classification. Experiments were conducted in order to compare linear and RBF kernels for the SVM classification stage in feature-level fusion (PD scenario) and decision-level fusion (PI scenario). Values for parameters $C = 1$ and $gamma = 0.01$ were determined by a grid search (for the C parameter considered values were 0.1, 0.3, 0.6, 1, 3, 6, 10, and for the *gamma* parameter considered values were 0.01, 0.03, 0.06, 0.1, 0.3, 0.6). The results are provided in Table 3.7. Using the RBF kernel did not provide an improvement in any of the considered cases.

Table 3.7: Recognition accuracy (%) for the FFD dataset using the proposed feature selection method and fusion of multiple feature sets for two SVM kernels: linear and RBF.

	SVM LIN	SVM RBF
PD (feature-level fusion)		
LTI + JD + JMHC	**98.67**	97.12
LTI + JD + JMHC + Acc	**98.67**	98.02
PI (decision-level fusion)		
LTI + JD + JMHC	**86.31**	85.74
LTI + JD + JMHC + Acc	**85.66**	84.92

State-of-the-art algorithms for action recognition were run on the FFD dataset in order to compare them with the proposed method. The results are shown in Table 3.8. The proposed method (86.31% in the PI scenario) clearly outperforms the other ones (76.14% for LOP/FTP in the PI scenario). Such a significant difference indicates that the state-of-the-art methods are not sufficient for the presented dataset, as they were designed for typical action recognition tasks, rather than for distinguishing similar actions based on their dynamics.

Table 3.8: Recognition accuracy (%) for the FFD dataset - the proposed method compared to state-of-the-art methods.

Method	PD	PI
EigenJoints [299]	35.04	29.89
MHI [23]	88.60	61.25
SkeletonNet [131]	93.12	64.36
C3D [261]	94.55	67.63
HON4D [210]	93.22	75.87
LOP/FTP [276]	94.21	76.14
LTI + JD + JMHC	**98.67**	**86.31**
LTI + JD + JMHC + Acc	**98.67**	**85.66**

Confusion matrices for the PD and PI scenarios are provided in Tables 3.9 and 3.10 respectively. It can be observed that two step actions, which are all considerably different, are easily recognized, even in the PI scenario. On the other hand, the four lunge actions are much more difficult to classify. Rapid (R) and jumping-sliding (JS) lunges are easier to recognize, as they have some distinctive characteristics - the former is the fastest and shortest and the latter is the longest and includes the foot sliding motion. The most difficult to recognize is the increasing speed (IS) lunge, due to the fact that its distinguishing feature is continuous acceleration. Since acceleration occurs in all of the considered actions, the difference between the IS and the other actions is very subtle. The distinctive characteristic of the lunge with waiting (WW) is sudden acceleration after a pause, which can be easily confused with the IS action, as in both cases the lunge is slower at first, and faster later on.

Table 3.9: Confusion matrix for the FFD dataset, PD scenario, using the proposed method: LTI + JD + JMHC, with feature-level fusion.

	R	IS	WW	JS	SF	SB
R	**99**	1	-	-	-	-
IS	-	**96**	4	-	-	-
WW	-	2	**98**	-	-	-
JS	-	1	-	**99**	-	-
SF	-	-	-	-	**100**	-
SB	-	-	-	-	-	**100**

Table 3.10: Confusion matrix for the FFD dataset, PI scenario, using the proposed method: LTI + JD + JMHC, with decision-level fusion.

	R	IS	WW	JS	SF	SB
R	**85.27**	12	1.82	0.91	-	-
IS	10.99	**71.64**	5.55	11.82	-	-
WW	4.55	18.18	**77.27**	-	-	-
JS	-	13.64	-	**86.36**	-	-
SF	-	-	-	-	**100**	-
SB	-	-	-	-	2.68	**97.32**

3.3.3 Results on UTD-MHAD dataset

The UTD-MHAD dataset differs from the FFD dataset in several aspects. There are significantly more actions (27), which include the motion of all joints. Also, not only 2D, but 3D motion is relevant for classification in this case. On the other hand, the actions are considerably different from each other, and are therefore much easier to recognize. LTI features, which were designed specifically to address the difference in the dynamics of similar actions, although useful for the FFD dataset, performed poorly on the UTD-MHAD dataset, probably due to the different specifics of actions, as well as a larger number of actions and considered joints. Therefore, LTI features were not used for this dataset.

In order to handle 3D motion, JD and JMHC features were computed separately in 3 orthogonal planes (*xy, xz, yz*). The proposed feature selection method was compared to state-of-the-art selection methods for each computed feature set. Experiments with different numbers of selected features were run for each set in order to determine the best size of the feature subset in each case. The results are presented in Table 3.11. In the case of JD features the proposed method outperformed the other ones in all 3 planes, although AdaBoost provided similar results in two cases. For the JMHC features, the proposed method was superior for the *xz* plane. In the other two planes, the proposed method was significantly better than the other feature selection methods, although dimensionality reduction with PCA proved to be more effective in these cases. Also, for the Acc features, PCA was superior.

Based on the experiments with the FFD dataset, decision-level fusion was employed, as it was much more effective when actions of previously unseen persons are considered. As in the case of the FFD dataset, employing Acc features is considered separately, as it includes data from an additional device. The results are provided in Table 3.12, with respect to all employed feature selection methods. The selection was run on all 7 feature sets separately, with features chosen in accordance with the previous experiments. Separate SVMs were applied to each feature set and the SVM outputs provided the input for the MLP classifier. MLP used one hidden layer, while the other parameters were chosen based on the grid search: number of neurons (6 to 27 with step 3, selected value 27), learning rate (0.1 to 0.6 with step 0.1, selected value 0.2), and momentum (0.1 to 0.4 with step 0.1, selected value 0.3). The proposed feature selection method proved to be superior. Including the Acc features provided for significantly better accuracy.

Table 3.11: Recognition accuracy (%) for the UTD-MHAD dataset using linear SVM and different feature selection methods on different feature sets.

		w/o sel.	PCA	AdaBoost	Lasso	Proposed
JD_{xy}	#feat	2640	100	500	900	600
	acc.	82.79	85.58	84.88	85.81	**86.05**
JD_{xz}	#feat	2640	200	1000	900	700
	acc.	84.19	75.81	**85.58**	85.35	**85.58**
JD_{yz}	#feat	2640	100	2000	1750	200
	acc.	83.95	79.07	**86.98**	85.12	**86.98**
$JMHC_{xy}$	#feat	16800	400	750	2000	2000
	acc.	63.02	**80**	66.51	72.56	78.14
$JMHC_{xz}$	#feat	16800	400	1250	600	800
	acc.	63.02	73.95	68.14	74.19	**82.09**
$JMHC_{yz}$	#feat	16800	400	1250	2500	4500
	acc.	64.19	**73.95**	65.12	68.84	71.63
Acc	#feat	168	100	120	140	150
	acc.	78.14	**79.53**	77.91	79.07	78.6

Table 3.12: Recognition accuracy (%) for the UTD-MHAD dataset using a decision-level fusion of multiple feature sets with respect to different feature selection methods. Feature selections were made for each feature set separately.

	w/o sel.	PCA	AdaBoost	Lasso	Proposed
JD + JMHC	90.89	92.29	91.12	91.82	**92.76**
JD + JMHC + Acc	90.89	93.39	92.99	92.52	**94.39**

As the final tuning step of the classification process, employing the RBF kernel during the SVM stage was evaluated. The results are provided in Table 3.13. Using SVM with the RBF kernel, and with parameters $C = 10$ and $gamma = 0.03$, improved the final accuracy in both considered cases (with and without Acc features).

Table 3.13: Recognition accuracy (%) for the UTD-MHAD dataset using the proposed feature selection method and decision-level fusion of multiple feature sets for two SVM kernels: linear and RBF.

	SVM LIN	SVM RBF
JD + JMHC	92.76	**93.93**
JD + JMHC + Acc	94.39	**94.91**

A comparison of the proposed method with state-of-the-art algorithms is shown in Table 3.14, including the employed modalities. The proposed method outperforms all other methods based on the same modalities (depth, skeleton and inertial), and the only, slightly better algorithm is VGG-16, which employs RGB data. It is worth noting, that in many scenarios, particularly in sports, it is beneficial not to use RGB data, as it raises

issues concerning lighting conditions, background noise and privacy. The confusion matrix is provided in Table 3.15. The results indicate that several actions are considerably more difficult to recognize, as 21 out of 27 actions are classified with 100% accuracy. Draw triangle and draw circle counter-clockwise are often misclassified, particularly the second one is easily mistaken with draw circle clockwise (in 43.75%). Other misclassified actions include: catch (25% recognized as knock), jog (25% recognized as walk), throw (18.75% recognized as swipe left), knock (6.25% recognized as wave).

Table 3.14: Recognition accuracy (%) for the UTD-MHAD dataset - the proposed method compared to state-of-the-art methods.

Method	Acc. (%)	Modalities
DMM-CRC [38]	79.10	Depth + Inertial
GF + LF [79]	84.89	Depth + Skeleton
SD-SR [8]	86.12	Skeleton
JTM + CNN [280]	87.90	Skeleton
DMM-CT-HOG-LBP-EOH [29]	88.40	Depth
DMM-CRC-LOGP [39]	91.50	Depth + Skeleton + Inertial
TPM-LLC-BoA [76]	93.02	Skeleton
MDACC [74]	93.26	Depth + Skeleton + Inertial
VGG-F [134]	94.60	RGB + Depth + Skeleton
VGG-16 [134]	95.11	RGB + Depth + Skeleton
JD + JMHC	**93.93**	**Depth + Skeleton**
JD + JMHC + Acc	**94.91**	**Depth + Skeleton + Inertial**

3.4 Summary

This chapter addressed the problem of distinguishing similar actions in sports, based on an analysis of action dynamics. A number of novel feature extraction methods were proposed - joint dynamics, local trace images, joint motion history context, acc features - which employ multiple modalities, namely depth maps and skeleton models acquired with the Kinect, as well as accelerometric data acquired with an IMU. A new feature selection algorithm was also introduced, capable of efficiently finding subsets, effective at distinguishing classes in data while minimizing feature redundancy. Both feature-level and decision-level fusion were employed in order to effectively combine information from multiple feature sets. A dedicated dataset with fencing footwork actions was recorded. It includes four similar lunge actions, and therefore allows to properly evaluate the proposed methods. Extensive experiments were conducted on the proposed dataset as well as the publicly available UTD-MHAD dataset.

The obtained results indicate that the proposed methods are significantly more effective for distinguishing between similar actions in the FFD dataset than state-of-the-art algorithms designed for general action recognition. This confirms that sports actions differ considerably from general actions and therefore require dedicated analysis methods. On the other hand, a subset of the proposed methods proved to be efficient for the UTD-MHAD dataset, which indicates, that the proposed dynamics-based approach can also be useful in general action recognition.

Table 3.15: Confusion matrix for the proposed method (JD + JMHC + Acc) on the UTD-MHAD dataset.

	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10	a11	a12	a13	a14	a15	a16	a17	a18	a19	a20	a21	a22	a23	a24	a25	a26	a27
swipe left	**100**	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
swipe right	-	**100**	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
wave	-	-	**100**	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
clap	-	-	-	**100**	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
throw	18.75	-	-	-	**81.25**	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
arm cross	-	-	-	-	-	**100**	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
basketball shoot	-	-	-	-	-	-	**100**	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
draw x	-	-	-	-	-	-	-	**100**	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
draw circle CW	-	-	-	-	-	-	-	-	**100**	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
draw circle CCW	-	-	-	-	-	-	-	-	43.75	**56.25**	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
draw triangle	-	-	6.25	-	-	-	-	-	-	12.5	**81.25**	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
bowling	-	-	-	-	-	-	-	-	-	-	-	**100**	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
boxing	-	-	-	-	-	-	-	-	-	-	-	-	**100**	-	-	-	-	-	-	-	-	-	-	-	-	-	-
baseball swing	-	-	-	-	-	-	-	-	-	-	-	-	-	**100**	-	-	-	-	-	-	-	-	-	-	-	-	-
tennis swing	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**100**	-	-	-	-	-	-	-	-	-	-	-	-
arm curl	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**100**	-	-	-	-	-	-	-	-	-	-	-
tennis serve	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**100**	-	-	-	-	-	-	-	-	-	-
push	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**100**	-	-	-	-	-	-	-	-	-
knock	-	-	6.25	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**93.75**	-	-	-	-	-	-	-	-
catch	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	25	**75**	-	-	-	-	-	-	-
pickup&throw	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**100**	-	-	-	-	-	-
jog	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**75**	25	-	-	-	-
walk	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**100**	-	-	-	-
sit to stand	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**100**	-	-	-
stand to sit	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**100**	-	-
lunge	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**100**	-
squat	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	**100**

The ability to properly recognize actions in sports is important in two cases. Firstly, it allows to determine which actions an athlete has chosen to perform in a given situation, that is the basis of analyzing and improving their tactics. Secondly, distinctive models of actions can be employed to evaluate the correctness of the performance of these actions. In both scenarios, action recognition constitutes for feedback which can be useful in improving one's sports skills.

Chapter 4

Real-time detection and analysis of actions in fencing footwork

This chapter concerns the detection and analysis of actions in continuous recordings of fencing footwork training routines. Automatic temporal segmentation is an important aspect of sports data analysis, as manual segmentation cannot be used in systems operating in real-time. Two modalities are employed - skeleton estimates from the Kinect and inertial data from a custom-built system with two IMUs. A novel model-based signal filtering method is proposed, which enables the effective detection of actions, as well as the extraction of relevant qualitative parameters. Experiments are conducted on two datasets recorded specifically for this work. The experimental results indicate that the proposed method can provide useful real-time feedback for fencers and their coaches.

Most of the works devoted to motion analysis, particularly those that consider action recognition, operate on datasets with temporally pre-segmented data, where each action execution is provided as a separate sample [34, 309]. While this is convenient for the development of algorithms, it has little application in real-world scenarios, unless proper segmentation methods are introduced as well. The problem of temporal data segmentation is rarely discussed in the literature. In the field of automatic sports analysis, it has been addressed in disciplines involving cyclic motion, by detecting repetitive actions such as swimming strokes [212] or pommel horse circles [226]. In this work, the detection of actions in a continuous, non-cyclic training routine is considered. In fencing footwork training, the fencer moves forward and backwards in steps and performs actions such as a lunge or a dodge. These actions are performed either on the command of a coach, or at will. In either case, actions can occur at any moment, therefore the training routine is not cyclic. In this work, lunge detection is considered, as this is the most important action in fencing footwork.

Another aspect of the automatic study of motion which is also rarely discussed in the literature is the qualitative analysis of actions. In the aforementioned work considering the pommel horse routine [226], the timing of motion cycles is determined. The authors of [11] provide real-time feedback in rowing, table tennis, and biathlon, with the use of acceleration plots, visualization of ball impact positions and visualization of barrel motion respectively. In fencing, the qualitative analysis of lunge actions was performed by making use of stereophotogrammetry [92], cameras [194], as well as electromyography and high-speed cameras [26]. However, none of these solutions provide real-time

feedback. In this work, lunge actions are recognized and analyzed immediately, during practice. Two approaches are proposed, based on skeleton data from the Kinect and inertial data from two IMUs. Once a lunge is detected, a number of qualitative parameters is calculated, namely duration, length, average and maximum speed, and acceleration, as well as hand timing. Feedback is provided in real-time, as the presented system is able to detect and analyze the lunge actions instantaneously and send the results wirelessly to a smartphone, which can be used during training. Therefore, fencers can correct their actions during practice, rather than receive feedback only afterwards. The proposed methods were published in [175].

This chapter is organized as follows. Section 4.1 discusses the data acquisition process with the Kinect and with a custom two IMU-system. In Section 4.2, new Kinect-based methods for detection and analysis of lunge actions are introduced. In Section 4.2.5, the proposed methods are adapted for IMU-based signals. Section 4.3 includes details regarding the experiments and discussion of the results. A summary is given in Section 4.4.

4.1 Data acquisition

The proposed methods for the detection and analysis of fencing footwork are based on two data modalities - skeleton data from the Kinect and inertial signals from two IMUs. Two datasets were acquired. In the first stage, the Kinect sensor was employed in order to verify the suitability of employing joint position data for the temporal segmentation of continuous recordings of fencing footwork practice. The first dataset was acquired solely with the Kinect. The sensor captured the side-view of the subject from approx. 3 meters distance. Each person was asked to perform a basic footwork training routine consisting of moving forward and backward with steps and performing lunge actions at will. Skeleton and depth data were recorded, the former as input for the developed algorithms, and the latter for the purpose of preparing the ground truth by way of manual labeling. Eight advanced fencers participated in data acquisition. Some of them were recorded twice (on different days). For each fencer, 2 to 5 recordings were captured, each including approx. 5 lunge actions. In total, the first dataset includes 31 recordings of 149 lunge actions. Each recording was manually labeled with labels indicating the start and end frames for each lunge action, as well as the frame in which the weapon arm is straightened. Manual labels were used as the ground truth in the experiments.

While the skeleton data provided by the Kinect proved to be efficient for the discussed task, the sensor itself is not quite convenient for use in training rooms. The practicing person needs a rather large space for exclusive use, as the sensor is placed at an approx. 3 meters distance and the space between the sensor and the person must be empty. Therefore, the next stage of research in this work considered employing IMU sensors, which are free of the aforementioned limitations. In order to capture both body motion and arm motion (which is an important qualitative parameter, as discussed in Section 4.2.3), two IMUs were employed. Professional IMUs are expensive, therefore a custom two-IMU system, more affordable for a typical fencing institution was designed, built and evaluated. The cost of the proposed system is approx. 10 times lower than typical commercial solutions.

The architecture of the assembled system is presented in Fig. 4.1 (left). It employs an Arduino Leonardo board with an Atmega32U4 microcontroller, a multiplexer which

enables connecting two I2C [120] devices simultaneously, and two Adafruit LSM9DS0 IMUs, with 9 degrees of freedom (3 axis measurements from accelerometer, magnetometer and gyroscope). The final device is presented in Fig. 4.1 (right). The data from the device were transferred via a long USB cable, which was sufficient for dataset acquisition, although in the next version, which is to be developed in the near future, the communication will be performed wirelessly via Bluetooth, which is obviously better in a practical scenario. The microcontroller was programmed to capture data from both IMUs simultaneously and provide raw readings from their accelerometers, gyroscopes, and magnetometers. The effective data delivery rate of the system is approx. 45 Hz.

The second dataset was acquired by employing both the Kinect and the assembled device. The first IMU was mounted near the elbow of the weapon-hand arm, in order to detect its forward movement. The second IMU was mounted on the chest, which was expected to be a good reference point for detecting lunge actions. The main objective was to capture inertial data, although data from the Kinect were recorded as well in order to compare both modalities. Once again, depth data were used for manual labeling. The synchronization of the skeleton and inertial data was performed by ensuring a simultaneous start and then resampling the inertial data to match the Kinect's recording frequency. Nine advanced fencers participated in the acquisition of the second dataset (five of whom were also present for the recording of the first dataset). For each person, 3 to 4 recordings were captured, each containing 5 to 6 lunge actions. The total size of the second dataset is 28 recordings of 162 lunges.

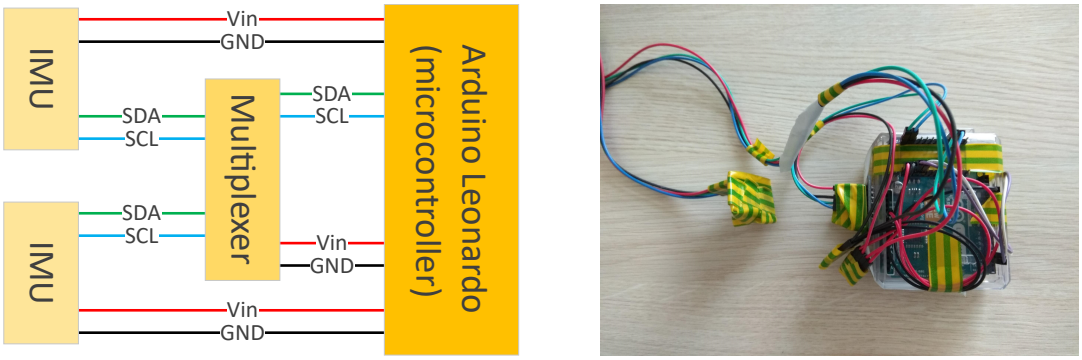

Fig. 4.1: Custom two-IMU system. Architecture (left) and the final device (right). Symbols: *Vin* - input power voltage, *GND* - ground, *SDA* - serial data line, *SCL* - serial clock line.

4.2 Methods

The workflow of detecting and analyzing actions in fencing footwork is as follows. Firstly, the signal segments which could potentially include lunge actions need to be detected. Secondly, such lunge action candidates need to be verified. Thirdly, the qualitative parameters should be extracted. This requires the key frames of the lunge action (start, end, arm straightening) to be detected precisely. Finally, feedback should be provided to the practicing subject in real-time. The proposed methods were initially designed for the data acquired with the Kinect. The detection, classification, qualitative analysis, and feedback are described in Sections 4.2.1, 4.2.2, 4.2.3, and 4.2.4 respectively. The adaptation of the proposed methods to IMU signals is discussed in Section 4.2.5.

4.2.1 Action detection

The proposed method for the detection of a lunge is based on the following characteristics of the action. A lunge is initiated by extending the front leg and dynamically pushing off with the back leg, followed by the front leg touching the ground, the fencer coming to a short halt, and then returning to the previous position by bending the knee of the back leg and taking the front leg back (see Fig. 4.2). It is worth noting, that arm extension is also an important part of the lunge action, and the timing between the body forward movement and arm straightening is a relevant qualitative parameter. The body motion can be analyzed in terms of velocity. At first, there is rapid acceleration, then the maximum velocity is reached during the forward movement, and then deceleration occurs just before coming to a halt, which results in zero velocity. When a fencer returns to the previous position, similar changes in velocity can be observed, but in the opposite direction. Therefore, the lunge action can be detected by finding a positive maximum peak in the velocity plot, followed by a negative minimum peak (see Fig. 4.3 left and right segments). The start of the action is determined by another minimum, occurring before the maximum peak. However, similar peaks are present when fencer performs a sequence consisting of a step forward and step backward (see Fig. 4.3 middle segment). Thus, the detection of a pair of positive and negative peaks results only in lunge candidates which need to be verified.

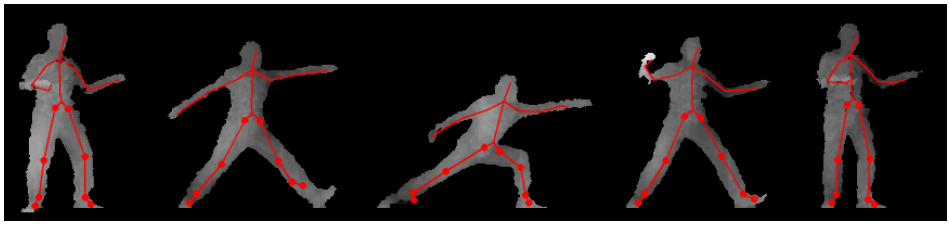

Fig. 4.2: Keyframes (depth and skeleton view) of lunge action. Starting from the fencing pose, the fencer extends the arm, then extends the front leg and pushes off with the back leg, then reaches the full extent of the lunge, then returns to the basic pose.

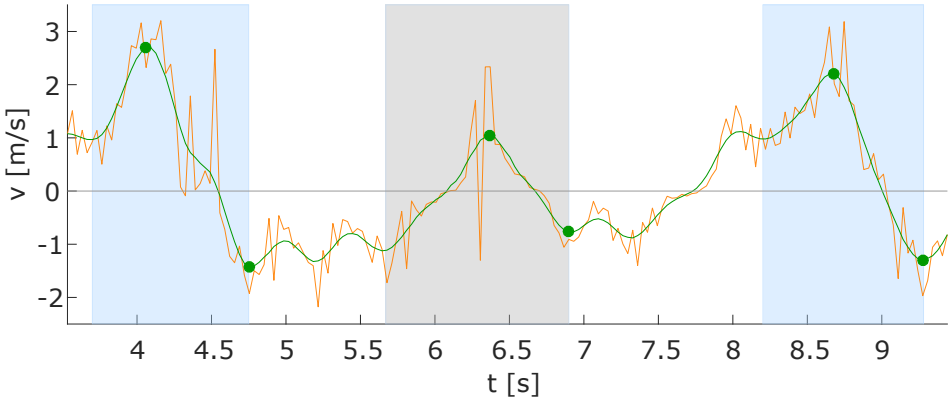

Fig. 4.3: Velocity over time of a fencer during footwork practice. Raw signal (orange line) is filtered (green line) in order to make signal analysis feasible. A sequence of a positive maximum peak and a negative minimum peak (green markers) indicates lunge candidates. There are three detected segments in the plot: left and right are actual lunges (blue background) and the middle one is a step forward and backward action (gray background).

The skeleton data provided by the Kinect includes 20 joint positions (see Fig. 3.6). Based on these positions, velocity v in each frame is calculated as the difference in positions p between two consecutive frames:

$$v(t) = p(t) - p(t-1) \tag{4.1}$$

In order to analyze the forward and backward movement of a fencer, the velocity of a single joint is sufficient. The joint corresponding to the spine base was chosen, as it is located near the center of the body, and therefore is tracked by the Kinect most consistently. Nevertheless, the raw signal is very noisy and needs to be filtered before applying algorithms for peak finding. An important criterion was to use filters, which enable local smoothing with a relatively low delay, so that the detection and analysis could be performed in real-time. The following filters were considered:

- Moving Average (MA) - the arithmetic average of velocity values v_i computed over a moving window of given length n:

$$ma = \frac{v_1 + v_2 + ... + v_n}{n} \tag{4.2}$$

  Due to significant noise in the Kinect skeleton data, a single pass of this filter was not sufficient, therefore a double pass was applied, with a shorter filter window in the second pass. Experimental results demonstrated, that double filtering with MA resulted in smooth signals.

- Locally Weighted Scatterplot Smooth (Loess) [49] - smoothing filter based on second degree polynomials, computed as follows. At first, in a window of length n, distances d_i are calculated between the data point v_c at the center of the window and the other data points v_i:

$$d_i = v_c - v_i, \quad i = 1, ..., n \tag{4.3}$$

  Next, based on these distances, weights w_i are computed:

$$w_i = (1 - |d_i|^3)^3, \quad i = 1, ..., n \tag{4.4}$$

  Then, a second degree polynomial is fitted, by using a weighted linear least-squares regression [180] to minimize the fit error e:

$$e = \sum_{i=1}^{n} w_i * (v_i - \hat{v}_i)^2. \tag{4.5}$$

  where $\hat{v}_i$ is the fitted value. The value of the fitted polynomial at the center of the window is used as the filtered value.

There are two goals when analyzing the velocity signal - the first is to find all relevant peaks, and the second is to avoid finding improper peaks, which can occur due to data noise. These goals are, in fact, contradictory. Strong smoothing removes improper peaks, but can result in removing some relevant peaks as well. Weak smoothing, on the other hand, preserves the relevant peaks, but can produce additional improper peaks. In the discussed scenario, it is important to find a minimum preceding the positive maximum peak, as it indicates the time when the lunge starts (see Fig. 4.4), which is crucial

for performing qualitative analysis (see Section 4.2.3). When fencers perform the lunge after stepping forward, they slow down only slightly, therefore the minimum is small and can be lost when too much filtering is applied (see Fig. 4.4 left). On the other hand, the Kinect has difficulties with properly tracking the joints during the slowing-down stage of the lunge, which sometimes results in outliers in the data that may be misidentified as peaks if the filtering is not sufficient (see Fig. 4.4 middle).

Experiments were conducted using both MA and Loess filters, at different window sizes, in order to verify how well-suited they are for this task. It was concluded, that the MA filter tends to produce smoother plots, therefore sometimes discarding the initial minimum (see Fig. 4.4 left), while the Loess filter tends to preserve peaks, although it generates improper peaks when outliers occur in the raw signal (see Fig. 4.4 middle). It is worth noting that there exists a modified version of the Loess filter dedicated to removing outliers [49], although it works well only when supplied with long signal segments, and is therefore not suitable for real-time filtering.

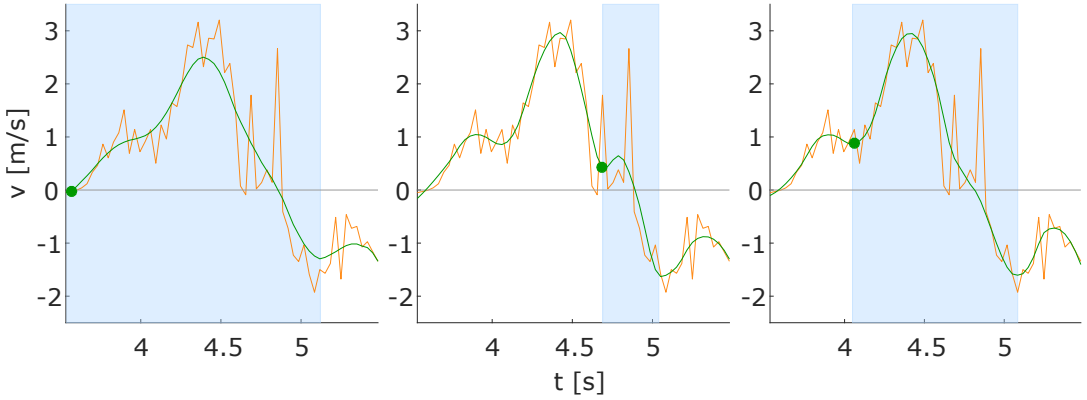

Fig. 4.4: Velocity over time for lunge action. Raw signal (red line) is filtered (green line) with different methods, which results in different detection of the start point (green marker) of action segment (blue background). The MA filter (left) does not preserve minimum on the rising slope, and the Loess filter (middle) produces improper peak on falling slope. The proposed method (right) correctly handles both slopes.

In order to address the discussed issues, a novel filtering method is proposed, which allows to take advantage of both the MA and Loess algorithms. The novelty consists in providing adaptive filtering which is based on known model of the analyzed signal. In this scenario, the model assumes that peaks should be preserved on the rising slope and smoothed on the falling slope. The proposed filter adapts to the signal by computing the filtered value as a weighted sum of the values from both the MA and Loess filters, with weights calculated on the basis of the current slope of the signal. Slope direction s is approximated with the first derivative of the MA-filtered signal $\hat{v}$ in a window of size n (equal to the window employed in the filter):

$$s = \sum_{i=1}^{n-1} \frac{\hat{v}_{i+1} - \hat{v}_i}{n-1}. \tag{4.6}$$

Slope direction s is then cut to range (-1,1):

$$s = \max(\min(s, 1), -1) \tag{4.7}$$

and normalized to range (0,1):

$$s = \frac{s+1}{2} \tag{4.8}$$

Therefore, $s < 0.5$ corresponds to falling slope and $s > 0.5$ corresponds to rising slope. Slope direction is employed directly to calculate the weights for the MA and Loess filters:

$$w_{movingavg} = 1 - s \tag{4.9}$$

$$w_{loess} = s \tag{4.10}$$

Since the slope is computed based on a filtered signal, it changes smoothly, therefore the transition between the MA and Loess values is also smooth. The proposed method results in the signal filtering being more similar to the Loess filter on the rising slope (see Fig. 4.4 middle and right), and to the MA filter on the falling slope (see Fig. 4.4 left and right).

The segment of interest detection in the signal is performed as follows. Peaks are identified as local maxima or minima in the filtered signal. When a pair of peaks of a maximum higher than zero followed by a minimum lower than zero is detected, it is marked as a lunge candidate, after which the key frames of the detected action segment are calculated. The start frame is found by going left from the maximum, looking for a point in which there is either a local minimum or a negative value, whichever occurs first. This corresponds to the start of the forward motion during the lunge. The end frame is found by going right from the maximum and looking for the first point with a negative value. This corresponds to the end of the forward motion. Returning to the basic pose is not analyzed and therefore it is not considered a part of the action. The frame where the arm is straightened is found by going left from the end frame and looking for a frame in which the angle formed by three of the joints: wrist, elbow and shoulder, changes from below to above of 160 degrees.

4.2.2 Action classification

The proposed method for action detection results in signal segments which correspond to either a lunge or a sequence of stepping forward and backward. Therefore, once an action segment is detected, classification is performed. Even though the velocity pattern of the considered joint (spine base) is similar for both actions, the overall motion is significantly different. Therefore, information regarding all tracked joints is employed.

Firstly, segments during which the arm is not straightened at all (based on the wrist-elbow-shoulder angle), are identified as non-lunge, as this is crucial for an action to be considered a lunge. The remaining segments are classified by using an algorithm for lunge action recognition that is based on the feature extraction and classification methods presented in Chapter 3. The x and y direction velocities are computed for each joint and then interpolated to a common length. Since the Kinect operates at 30 Hz and the average duration of an extracted action segment is approx. 1 second, the target length is equal to 32 samples. The velocities are transformed to the frequency domain by applying the FFT. The first 3 coefficients of the FFT, computed for each joint, in x and y direction, are used as features. Using 3 coefficients, 20 joints and 2 directions, the total size of a feature vector is 120. Linear SVM is then used for classification (see Section 4.3 for details).

4.2.3 Qualitative analysis

Once a lunge action is detected alongside its key frames, qualitative analysis can be performed. The following parameters are calculated, as they were considered by fencing coaches as important:

- Hand timing - a crucial aspect of properly executing a lunge is to start by extending the weapon hand and only then moving the whole body forward. Otherwise, the opponent is likely to perform an effective counteraction. Therefore, the time difference between the arm straightening and the start of the forward motion is determined.
- Duration - a faster lunge gives the opponent less time to react. Duration is calculated as the time between the start and end frames of the detected action segment.
- Length - a longer lunge allows to attack from a greater distance. Lunge length is determined as the difference between the extreme positions of the selected body joint (spine base) during the action.
- Average and maximum acceleration - a dynamic lunge results in a greater length and shorter duration, therefore the dynamic parameters of the motion are relevant. The average and maximum acceleration are computed as the mean and maximum change of velocity on the rising slope of the filtered signal respectively.
- Average and maximum velocity - similarly to acceleration, velocity concerns measuring the dynamics of the motion. The velocity parameters are computed as the mean and maximum value on the rising slope of the filtered signal, respectively.

Fencers can plan their practice according to these parameters, either by choosing a single parameter to improve at a time, or multiple, which is more difficult, but important for advanced athletes.

4.2.4 Feedback

Sport support systems are much more effective when feedback is provided in real-time, as the practicing person can correct their movement while performing exercises. Several possibilities for providing feedback were considered. The data from the Kinect are processed on a laptop, therefore the parameters of the detected lunge actions can be displayed on the laptop screen. However, it is difficult to make the laptop easily visible during the entire practice, as the fencer moves forward and backward and thus changes their position a lot. Therefore, another device was employed for providing feedback, namely a smartphone. Since the footwork is usually practiced without the weapon, the fencer can hold a smartphone in their hand, or it can be placed in a transparent case attached to the forearm. Once a lunge is detected and analyzed, the computer transmits the calculated parameters wirelessly to be displayed in a dedicated application on the mobile device. This is a simple, yet effective solution which does not require any additional devices as the fencers can use their own smartphones. The feedback is provided instantaneously. Including processing and transfer to the mobile device, the feedback is available in less than half a second after finishing the forward motion part of the lunge action. Since motion parameters are evaluated in real-time, corrections in the movement can be made during the practice session.

While real-time feedback is most important, post-practice analysis is also interesting, as it provides additional information. The practice can be recorded with the developed software and the fencer or their coach can analyze it afterwards. The software enables replaying the recording, by showing the depth, silhouette and skeleton views, plotting the raw and filtered velocity signal in a selected time window, highlighting detected lunge segments and presenting their qualitative analysis (see Fig. 4.5). Statistics from the entire recording are computed as well. It is worth noting that since the system analyzes the signal and displays information in real-time, in can also be used by the coach for overseeing a practicing person.

Fig. 4.5: System for footwork practice analysis.

Finally, two other options of delivering real-time feedback were considered and may be employed in the future. The first option is an augmented reality (AR) headset (see Chapter 5), which could display the parameters of the analyzed lunge on semi-transparent glasses. This solution seems to be more convenient than a smartphone attached to the hand, although it requires the AR headset which is very expensive. Another option would be a smartwatch, which could be mounted on the wrist, similarly to the smartphone, but at a significantly lower weight. Although basic smartwatches are not that expensive, they are still much less popular than smartphones and users may prefer using the smartphone that they already have rather than acquire a smartwatch.

4.2.5 IMU-based methods

Signals from IMUs (accelerometric, angular velocity, and magnetic) provide a considerably different type of information than the visual data from the Kinect. Therefore, the

proposed methods needed to be modified in order to produce similar feedback. The idea was to detect the lunge and arm straightening with chest-mounted and elbow-mounted sensors, respectively. Lunge detection is based on accelerometric signal. It is possible to obtain the velocity signal by integrating the acceleration, although the accumulation of error makes this approach impractical. Therefore, acceleration in the horizontal direction is used. Contrary to the Kinect data, peak analysis in the acceleration signal is not sufficient to detect potential lunge segments. IMUs are sensitive to the dynamics of the motion, and therefore, without additional information, it is not possible to efficiently identify which peaks correspond to a lunge action and which correspond to other actions or changes in speed when performing the exercise (see blue line in Fig. 4.6). For this reason, the lunge segments are identified by first detecting a moment in which the arm is straightened by using the elbow-mounted sensor, and only then finding the exact start and end points of the action by using the accelerometer signal from the chest-mounted sensor. A drawback of this approach is that some actions, such as moving forward and backward with a straightened arm, are identified as a lunge as well. Although typically fencers straighten the arm only during the lunge, it is not always the case, therefore this is a relevant limitation.

Various approaches can be used for detecting arm straightening. The most obvious one would be to use sensor fusion algorithms to compute the orientation [165]. However, initial experiments showed that these either introduce a significant delay or produce a noisy estimation of angles, and are therefore not suitable for this application. Another approach which was under consideration was to employ gyroscope readings, which should generate a peak at such a motion. However, the data was too noisy, as the sensor was very sensitive to the motion of the arm resulting from the overall movement of the body. Similar problems were present when using accelerometer data. Therefore, arm straightening is detected by using magnetometer readings captured along the axis aligned with the arm. The magnetometer produces much more stable data and arm-straightening is easily detected by finding a raising slope with a height above a given threshold, which is determined experimentally (see purple line in Fig. 4.6).

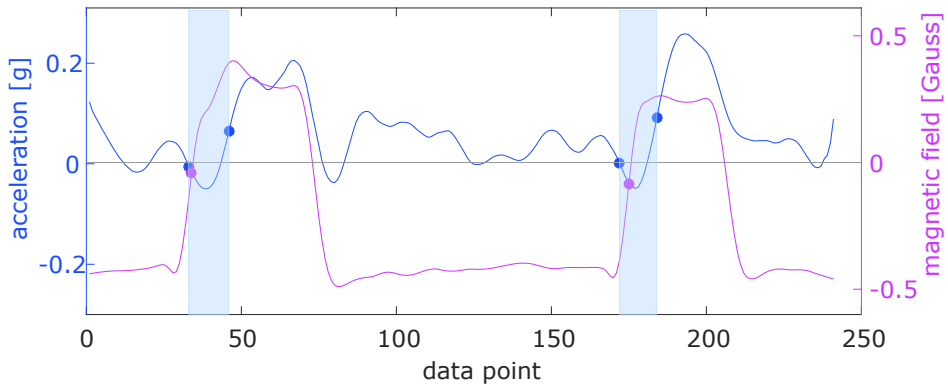

Fig. 4.6: Detection of lunge action using signals from two inertial sensors. Purple line represents filtered magnetometer readings from elbow-mounted sensor and blue line represents filtered accelerometer signal from chest-mounted sensor. Markers indicate detected keypoints and blue backgrounds indicate detected segments.

Once the arm straightening is detected, the algorithm waits for another 30 data points to arrive, and then begins searching for the keypoints of the action. The exact point of arm straightening is determined by finding where the rising slope of the filtered mag-

netometer signal is the steepest, i.e. where the difference in value in the filtered signal between the current and the preceding point is at its highest (see purple markers in Fig. 4.6). The end point is determined as the first positive peak in the filtered accelerometer signal (chest-mounted sensor) after the arm-straightening (see blue markers in Fig. 4.6). This corresponds to the time, when the forward motion stops. The start point is identified as the middle of the first falling slope preceding the peak which indicates the end point (see blue markers in Fig. 4.6). This corresponds to the moment when the forward motion starts.

No additional classification is performed for the detected action segment, as the available data is not sufficient to provide more relevant information for this task. It might be beneficial to mount another sensor on the knee and employ classification based on methods presented in Chapter 3.2.1, however, only two sensors were available during the experiments. The qualitative parameters provided by the IMU-based method are slightly limited in comparison with the Kinect-based algorithm. Hand timing and lunge duration are computed based on the detected key frames. Average and maximum acceleration are provided based on the filtered acceleration signal. Average and maximum velocity, as well as lunge length are not computed, as the accuracy would be too low due to the accumulation of error. The feedback is provided in a similar manner as in the Kinect-based method. It is worth noting, however, that with inertial sensors, it is possible to eliminate the need for a computer, as the sensors could communicate directly with a smartphone or smartwatch, which would handle the processing.

4.3 Experiments and results

The proposed methods were verified on two datasets recorded specifically for this work (see Section 4.1), as well as in experiments with fencing coaches. The first dataset includes data only from the Kinect, while the second dataset includes data from both the Kinect and the IMUs, allowing for a comparison of the two approaches. The ground truth was obtained for both datasets by manually labeling the recordings, including the key frames (start, end, arm-straightening) for each lunge, based on the depth data from the Kinect. The experiments involved employing 3 different filters: MA, Loess, and the proposed model-based method, with a number of different filter window lengths, as provided in the result tables. The MA filter employed double filtering, therefore two values are given for this filter.

Both the quantitative and qualitative parameters of the detection were measured. The number of all detected action segments is provided, as well as the number of those identified as a lunge. The true positive (TP) parameter represents how many of the segments classified as a lunge matched a lunge segment in the ground truth. Segments are considered as matching if the middle of the detected segment lays between the start and end of the actual action segment. The false positive (FP) parameter represents how many of the segments classified as a lunge were not matching any of the lunge actions from the ground truth. Additionally, the number of poorly detected segments (Poor) is given. A segment is considered poorly detected when the difference between the detected and ground truth segments (frame-wise) is greater than their common part. Based on these parameters, the performance measurements are computed. Recall represents the fraction of the actual lunges that were detected. Precision represents the fraction of detected lunges that were actual lunges. The formulas for the employed parameters are as follows [224]:

$$TP = \#\{a \in A \mid a \textit{ is an actual lunge}\} \tag{4.11}$$

$$FP = \#\{a \in A \mid a \textit{ is not an actual lunge}\} \tag{4.12}$$

$$Recall = \frac{TP}{N} \tag{4.13}$$

$$Precision = \frac{TP}{L} \tag{4.14}$$

where: TP - true positive, FP - false positive, a - detected action segment, A - set of all detected actions segments, N - number of lunge actions in ground truth, L - number of detected action segments classified as lunge action.

For the Kinect-based method, the classification stage includes employing SVM. In the experiments, a separate SVM model was trained for each case, defined by a unique set of arguments: dataset, person, filter type, filter length. There were 8 persons in the first dataset and 9 in the second. For each person, the SVM model was trained on all other persons in the considered dataset, resulting in leave-one-out cross-validation [139]. Weka [289] implementation of linear SVM was used, which provided automatic data normalization. Setting the parameter $C = 1$ was sufficient to obtain 100% recall and precision, therefore no experiments with other kernels and parameters were conducted.

The evaluation of the qualitative analysis of the detected lunge segments was performed by computing the mean absolute difference between the key frames (start, end, arm-straightening) from the detected segments and from the ground truth. The presented results also include the standard deviation. Differences between the key frames are denoted as direct parameters, as they are computed directly from the data, but not presented to the user. Based on that, the indirect parameters are computed, which constitute actual feedback for the fencers.

The accuracy of determining the three indirect parameters is verified: hand timing (the difference between lunge start and arm-straightening), lunge duration, and lunge length. The mean absolute difference of these parameters between the detected segments and their corresponding ground truth segments is provided, along with the standard deviation. Timing and duration are provided in frames, and the length is given in centimeters, as the Kinect uses such units for its skeleton data. The actual ground truth for the length was not acquired, therefore the results for this parameter represent only how the accuracy of key frame detection corresponds to the length estimation, based on the Kinect's position estimation. The accuracy of other parameters provided as feedback, acceleration and velocity, is not evaluated, as they do not depend on the exact positions of the key frames and no ground truth was available for them. According to fencing coaches, the most important of the discussed parameters is hand timing, it was therefore used as the criteria for marking the best filter length in the qualitative results.

4.3.1 Dataset 1

The results for Kinect-based lunge detection and the analysis for dataset 1 are presented in Tables 4.1 and 4.2 respectively. For the detection (see Table 4.1), a 100% recall was obtained for all filters: MA with a length (7;3), Loess with lengths of 17 and 19, pro-

posed with lengths of 13, 15 and 17. However, only the proposed method provided at the same time 100% precision, for filter lengths 13 and 17. The number of poorly detected segments was similar for the Loess and the proposed filter (0 in the best case for both) and higher for the MA filter (2 in the best case).

The results for the lunge action analysis (see Table. 4.2) show, that in regard to the parameters computed directly, the start frame was the most difficult to determine precisely (best accuracy: 1.27 ± 1.03 frames), while the end and arm-straightening frames were easier to ascertain (best accuracies: 0.62 ± 0.60 and 0.52 ± 0.61 frames, respectively). The proposed method was superior in detecting the key frames, particularly for the start frame. As for the indirect parameters, hand timing and lunge duration were detected with similar accuracy: 1.53 ± 1.03 and 1.64 ± 1.25 frames respectively. Lunge length accuracy in the best case was 3.73 ± 3.71 centimeters, although this result does not account for errors in the measurements introduced by the Kinect itself. The proposed method, with a filter length of 15, provided the best accuracy for all of the indirect parameters.

Table 4.1: Evaluation of lunge action detection, first dataset, Kinect-based method. Number of detected segments is given for: all detected action segments (Total); those classified as lunge (Lunge); true positive (TP); false positive (FP); poorly detected segments (Poor). Ground truth contains 149 lunge actions.

Filter	Number of detected segments					Performance (%)	
type, length	Total	Lunge	TP	FP	Poor	Recall	Precision
MA (5; 3)	207	149	148	1	3	99.33	99.33
MA (7; 3)	**205**	**150**	**149**	**1**	**2**	**100.00**	**99.33**
MA (7; 5)	205	146	145	1	2	97.32	99.32
MA (9; 5)	204	146	145	1	4	97.32	99.32
MA (9; 7)	201	142	141	1	5	94.63	99.30
MA (11; 7)	199	127	125	2	6	83.89	98.43
Loess (13)	220	146	146	0	2	97.99	100.00
Loess (15)	216	147	147	0	1	98.66	100.00
Loess (17)	**210**	**150**	**149**	**1**	**0**	**100.00**	**99.33**
Loess (19)	**210**	**150**	**149**	**1**	**0**	**100.00**	**99.33**
Loess (21)	207	149	148	1	2	99.33	99.33
Loess (23)	204	147	146	1	3	97.99	99.32
Proposed (13)	204	149	149	0	1	100.00	100.00
Proposed (15)	205	150	149	1	0	100.00	99.33
Proposed (17)	**201**	**149**	**149**	**0**	**0**	**100.00**	**100.00**
Proposed (19)	200	147	146	1	1	97.99	99.32
Proposed (21)	200	149	148	1	2	99.33	99.33
Proposed (23)	200	144	143	1	3	95.97	99.31

Table 4.2: Evaluation of lunge action analysis, first dataset, Kinect-based method. Mean value and standard deviation are presented. Direct parameters: accuracy for detecting start frame, end frame, and arm-straightening, provided in frames. Indirect parameters (provided as feedback): hand timing and duration, given in frames; length, provided in centimeters. Filter lengths with best hand timing results are marked.

Filter	Direct parameters			Indirect parameters		
(type, length)	Start [f]	End [f]	Arm strght. [f]	Timing [f]	Duration [f]	Length [cm]
MA (5; 3)	**1.92 ± 1.72**	**0.85 ± 0.82**	**0.81 ± 0.98**	**1.77 ± 1.44**	**2.32 ± 1.95**	**6.16 ± 6.27**
MA (7; 3)	2.00 ± 2.20	0.76 ± 0.71	0.64 ± 0.77	2.07 ± 2.00	2.39 ± 2.43	6.35 ± 7.82
MA (7; 5)	2.17 ± 2.33	0.72 ± 0.69	0.60 ± 0.71	2.37 ± 2.23	2.54 ± 2.55	7.30 ± 8.51
MA (9; 5)	3.17 ± 2.79	0.65 ± 0.64	0.63 ± 0.73	3.30 ± 2.75	3.52 ± 3.05	10.65 ± 9.96
MA (9; 7)	3.73 ± 3.05	0.63 ± 0.60	0.64 ± 0.76	3.81 ± 3.03	4.10 ± 3.30	12.48 ± 10.86
MA (11; 7)	5.64 ± 3.75	0.65 ± 0.65	0.58 ± 0.70	5.64 ± 3.86	6.09 ± 4.02	18.36 ± 12.46
Loess (13)	2.11 ± 1.89	0.80 ± 0.70	0.70 ± 0.85	2.17 ± 1.84	2.44 ± 2.06	6.68 ± 6.51
Loess (15)	**1.45 ± 1.27**	**0.82 ± 0.75**	**0.61 ± 0.76**	**1.60 ± 1.26**	**1.87 ± 1.53**	**4.61 ± 4.75**
Loess (17)	1.42 ± 1.58	0.81 ± 0.78	0.52 ± 0.61	1.62 ± 1.57	1.89 ± 1.81	4.73 ± 5.75
Loess (19)	1.57 ± 1.72	0.77 ± 0.69	0.52 ± 0.61	1.74 ± 1.69	1.99 ± 1.85	5.35 ± 6.40
Loess (21)	2.05 ± 2.17	0.71 ± 0.64	0.57 ± 0.66	2.15 ± 2.18	2.37 ± 2.31	7.04 ± 7.95
Loess (23)	2.47 ± 2.25	0.70 ± 0.58	0.60 ± 0.70	2.57 ± 2.31	2.80 ± 2.42	8.43 ± 7.71
Proposed (13)	1.76 ± 1.48	0.62 ± 0.61	0.69 ± 0.86	1.87 ± 1.43	2.01 ± 1.77	5.18 ± 5.43
Proposed (15)	**1.27 ± 1.03**	**0.64 ± 0.63**	**0.54 ± 0.66**	**1.53 ± 1.03**	**1.64 ± 1.25**	**3.73 ± 3.71**
Proposed (17)	1.44 ± 1.52	0.62 ± 0.60	0.52 ± 0.61	1.68 ± 1.45	1.87 ± 1.66	4.66 ± 5.60
Proposed (19)	1.74 ± 1.98	0.66 ± 0.62	0.52 ± 0.61	1.94 ± 1.92	2.18 ± 2.08	5.73 ± 7.34
Proposed (21)	2.26 ± 2.16	0.66 ± 0.65	0.59 ± 0.71	2.39 ± 2.14	2.63 ± 2.44	7.57 ± 8.36
Proposed (23)	2.60 ± 2.29	0.72 ± 0.64	0.63 ± 0.74	2.73 ± 2.28	3.06 ± 2.57	8.82 ± 7.93

4.3.2 Dataset 2

The results for dataset 2 using the Kinect-based method are presented in Tables 4.3 and 4.4. The detection in the second dataset (see Table 4.3) proved to be slightly less difficult than the first one, as in several cases it was possible to obtain both 100% recall and 100% precision at the same time: MA (7;3) and (7;5); Loess 19, proposed 17, 19, and 21. Also, in the majority of cases there were no poorly detected segments. The proposed method proved to be the most flexible, as it provided error-free recognition at 3 different filter lengths.

Regarding lunge analysis (see Table 4.4), the best obtained accuracies for the direct parameters were as follows: start: 1.23 ± 1.12 frames (proposed 17), end: 0.60 ± 0.61 (Loess 19), arm-straightening: 0.60 ± 0.72 (MA (7;5) and proposed 13 and 15). For the indirect parameters: hand timing 1.45 ± 1.52 frames (proposed 19), duration: 1.40 ± 1.29 frames (Loess 19), length: 3.26 ± 3.29 centimeters (proposed 17). The proposed method provided superior accuracies for both timing and length, and was slightly worse than the Loess filter for calculating the duration of the lunge. The obtained accuracies were similar to the ones in dataset 1, although different filter lengths were optimal between datasets. The most useful filter seems to be the proposed method with a filter length of 17, as it provided error-less detection for both datasets, as well as close to the best accuracies for lunge analysis.

Table 4.3: Evaluation of lunge action detection, second dataset, Kinect-based method. Number of detected segments is given for: all detected action segments (Total); those classified as lunge (Lunge); true positive (TP); false positive (FP); poorly detected segments (Poor). Ground truth contains 162 lunge actions.

Filter	Number of detected segments					Performance (%)	
type, length	Total	Lunge	TP	FP	Poor	Recall	Precision
MA (5; 3)	260	164	161	3	2	99.38	98.17
MA (7; 3)	**250**	**162**	**162**	**0**	**0**	**100.00**	**100.00**
MA (7; 5)	**246**	**162**	**162**	**0**	**0**	**100.00**	**100.00**
MA (9; 5)	242	161	160	1	0	98.77	99.38
MA (9; 7)	239	160	160	0	0	98.77	100.00
MA (11; 7)	238	155	155	0	0	95.68	100.00
MA (11; 9)	238	150	150	0	2	92.59	100.00
Loess (13)	268	159	156	3	3	96.30	98.11
Loess (15)	263	162	159	3	0	98.15	98.15
Loess (17)	256	162	160	2	0	98.77	98.77
Loess (19)	**255**	**162**	**162**	**0**	**0**	**100.00**	**100.00**
Loess (21)	250	163	162	1	0	100.00	99.39
Loess (23)	249	163	162	1	0	100.00	99.39
Proposed (13)	249	163	162	1	0	100.00	99.39
Proposed (15)	246	163	162	1	0	100.00	99.39
Proposed (17)	**241**	**162**	**162**	**0**	**0**	**100.00**	**100.00**
Proposed (19)	**239**	**162**	**162**	**0**	**0**	**100.00**	**100.00**
Proposed (21)	**239**	**162**	**162**	**0**	**0**	**100.00**	**100.00**
Proposed (23)	239	162	161	1	0	99.38	99.38

The results for dataset 2 using the IMU-based method are presented in Tables 4.5 and 4.6. The data from the IMUs were originally acquired at 45 Hz, but then re-sampled to 30 Hz, to match the Kinect data per-frame. Since the re-sampled IMU data points correspond to Kinect frames, the results for the IMU-based method are presented in frames as well. Synchronization was performed by starting and finishing the recordings from both the Kinect and the IMUs at the same time. Although initially this approach seemed sufficient, further analysis showed that in some of the recordings the start times for both devices did not match exactly, due to different initialization times, therefore the synchronization is not always accurate. This issue is relevant mostly for the evaluation of the direct parameters of the lunge analysis. The calculation of the indirect parameters cancels out the time shift between both modalities, as it uses relative frame distances. Also, synchronization discrepancies are small enough to not impact the performance evaluation of the lunge action detection process except for the number of poorly detected segments, which, for this reason, is not presented in this case.

Table 4.4: Evaluation of lunge action analysis, second dataset, Kinect-based method. Mean value and standard deviation are presented. Direct parameters: accuracy for detecting start frame, end frame and arm-straightening, given in frames. Indirect parameters (provided as feedback): hand timing and duration, given in frames; length, given in centimeters. Filter lengths with best hand timing results are marked.

Filter	Direct parameters			Indirect parameters		
(type, length)	Start [f]	End [f]	Arm strght. [f]	Timing [f]	Duration [f]	Length [cm]
MA (5; 3)	1.63 ± 1.42	0.67 ± 0.74	0.62 ± 0.73	1.92 ± 1.65	1.70 ± 1.32	4.74 ± 5.40
MA (7; 3)	1.34 ± 1.08	0.60 ± 0.67	0.64 ± 0.79	1.62 ± 1.32	1.42 ± 1.09	3.75 ± 3.35
MA (7; 5)	**1.39 ± 1.24**	**0.61 ± 0.64**	**0.60 ± 0.72**	**1.58 ± 1.46**	**1.56 ± 1.27**	**3.66 ± 3.70**
MA (9; 5)	2.09 ± 2.24	0.66 ± 0.64	0.64 ± 0.78	2.26 ± 2.32	2.28 ± 2.35	6.04 ± 7.11
MA (9; 7)	2.96 ± 3.17	0.65 ± 0.62	0.61 ± 0.72	3.09 ± 3.17	3.21 ± 3.24	8.68 ± 10.29
MA (11; 7)	4.20 ± 3.79	0.67 ± 0.64	0.66 ± 0.77	4.25 ± 3.76	4.50 ± 3.92	12.20 ± 12.34
MA (11; 9)	5.53 ± 4.28	0.71 ± 0.63	0.72 ± 0.77	5.62 ± 4.04	5.87 ± 4.40	16.51 ± 14.06
Loess (13)	1.88 ± 1.60	0.75 ± 0.72	0.61 ± 0.73	2.15 ± 1.89	1.91 ± 1.55	5.82 ± 6.51
Loess (15)	1.48 ± 1.12	0.64 ± 0.71	0.62 ± 0.74	1.73 ± 1.36	1.48 ± 1.14	3.93 ± 3.80
Loess (17)	1.40 ± 1.23	0.64 ± 0.66	0.63 ± 0.78	1.69 ± 1.58	1.52 ± 1.33	3.60 ± 3.58
Loess (19)	**1.29 ± 1.21**	**0.60 ± 0.61**	**0.63 ± 0.78**	**1.56 ± 1.51**	**1.40 ± 1.29**	**3.39 ± 3.72**
Loess (21)	1.40 ± 1.29	0.57 ± 0.60	0.63 ± 0.78	1.64 ± 1.63	1.45 ± 1.45	3.96 ± 3.89
Loess (23)	1.64 ± 1.55	0.55 ± 0.61	0.63 ± 0.78	1.88 ± 1.85	1.70 ± 1.61	4.59 ± 4.78
Proposed (13)	1.40 ± 1.08	0.65 ± 0.63	0.60 ± 0.72	1.69 ± 1.38	1.53 ± 1.08	3.89 ± 3.76
Proposed (15)	1.31 ± 1.08	0.63 ± 0.62	0.60 ± 0.72	1.52 ± 1.34	1.48 ± 1.14	3.44 ± 3.40
Proposed (17)	1.23 ± 1.12	0.65 ± 0.62	0.63 ± 0.78	1.49 ± 1.45	1.52 ± 1.18	3.26 ± 3.29
Proposed (19)	**1.23 ± 1.17**	**0.66 ± 0.65**	**0.63 ± 0.78**	**1.45 ± 1.52**	**1.44 ± 1.21**	**3.29 ± 3.69**
Proposed (21)	1.68 ± 1.80	0.68 ± 0.61	0.63 ± 0.78	1.94 ± 2.09	1.95 ± 1.87	4.83 ± 5.66
Proposed (23)	1.79 ± 1.76	0.70 ± 0.63	0.63 ± 0.78	1.96 ± 2.03	2.09 ± 1.81	5.17 ± 5.48

The best detection performance for the IMU-based method (see Table 4.5) was 99.38% recall and 98.77% precision. Such results were obtained for all filters at multiple filter lengths). All detection errors resulted from the method depending on finding the arm-straightening moment as the basis of lunge segment detection. One actual lunge action was not detected in any case, due to the fact that the person had the arm in an already almost straightened position at the start of the recording. At least two false positives occurred in all cases, due to the person performing arm-straightening without performing a lunge.

The best lunge analysis accuracies for the IMU-based method (see Table 4.6), obtained for the direct parameters were: start: 3.13 ± 2.50 frames (Loess 21), end: 2.24 ± 1.59 frames (MA (11;9)), arm straightening: 2.44 ± 1.75 frames (proposed 23). Regarding indirect params, both best hand timing accuracy, 1.99 ± 2.02 frames, as well as best duration accuracy, 2.41 ± 2.04 frames, were provided by the proposed filter, with lengths of 19 and 21, respectively.

Compared to the Kinect-based method, the IMU-based method is slightly less efficient, in regard to all considered parameters. Nevertheless, even though it was unable to provide 100% recall or precision, errors occur very rarely, and can be avoided as long as the practicing person does not perform arm-straightening in actions other than the lunge. This is a limitation, albeit it is acceptable. The accuracy of lunge action analysis

is also lower than with the Kinect-based method, particularly for the direct parameters, although the evaluation of those is slightly hindered by the synchronization which is not always accurate, as well as the fact that the nature of video and inertial segments is different - during a dynamic motion, changes in acceleration do not always exactly match the changes in position. On the other hand, indirect parameters, which are the most relevant, as only they are provided as feedback to the user, are determined with a much better accuracy. The best hand timing for the IMU-based method is 1.99 ± 2.02 frames, which is only half a frame worse than the best result obtained with the Kinect-based method (1.45 ± 1.52 frames). In conclusion, the accuracy difference between the Kinect-based and IMU-based methods is not significant and should have little impact on the final usefulness of the system.

Table 4.5: Evaluation of lunge action detection, second dataset, IMU-based method. Number of detected segments is given for: all detected action segments (Total); those classified as lunge (Lunge); true positive (TP); false positive (FP). Ground truth contains 162 lunge actions.

Filter	Number of detected segments				Performance (%)	
type, length	Total	Lunge	TP	FP	Recall	Precision
MA (5; 3)	164	164	158	6	97.53	96.34
MA (7; 3)	163	163	160	3	98.77	98.16
MA (7; 5)	163	163	160	3	98.77	98.16
MA (9; 5)	**163**	**163**	**161**	**2**	**99.38**	**98.77**
MA (9; 7)	**163**	**163**	**161**	**2**	**99.38**	**98.77**
MA (11; 7)	**163**	**163**	**161**	**2**	**99.38**	**98.77**
MA (11; 9)	**163**	**163**	**161**	**2**	**99.38**	**98.77**
Loess (13)	164	164	156	8	96.30	95.12
Loess (15)	164	164	157	7	96.91	95.73
Loess (17)	164	164	158	6	97.53	96.34
Loess (19)	**164**	**164**	**161**	**3**	**99.38**	**98.17**
Loess (21)	**164**	**164**	**161**	**3**	**99.38**	**98.17**
Loess (23)	**164**	**164**	**161**	**3**	**99.38**	**98.17**
Proposed (13)	163	163	160	3	98.77	98.16
Proposed (15)	162	162	158	4	97.53	97.53
Proposed (17)	**163**	**163**	**161**	**2**	**99.38**	**98.77**
Proposed (19)	**163**	**163**	**161**	**2**	**99.38**	**98.77**
Proposed (21)	**163**	**163**	**161**	**2**	**99.38**	**98.77**
Proposed (23)	**163**	**163**	**161**	**2**	**99.38**	**98.77**

Table 4.6: Evaluation of lunge action analysis, second dataset, IMU-based method. Mean value and standard deviation are presented. Direct parameters: accuracy for detecting start frame, end frame and arm-straightening, given in frames. Indirect parameters (provided as feedback): hand timing and duration, given in frames. Filter lengths with best hand timing results are marked.

Filter	Direct parameters			Indirect parameters	
(type, length)	Start [f]	End [f]	Arm strght. [f]	Timing [f]	Duration [f]
MA (5; 3)	4.11 ± 2.71	3.64 ± 2.27	2.77 ± 1.63	3.69 ± 2.58	4.85 ± 3.14
MA (7; 3)	3.55 ± 2.76	3.10 ± 1.93	2.70 ± 1.70	2.73 ± 2.12	3.58 ± 2.34
MA (7; 5)	3.32 ± 2.36	2.84 ± 1.78	2.79 ± 1.70	2.10 ± 1.76	2.87 ± 2.07
MA (9; 5)	3.31 ± 2.51	2.58 ± 1.60	2.72 ± 1.77	2.10 ± 1.80	2.65 ± 2.01
MA (9; 7)	**3.55 ± 2.65**	**2.44 ± 1.57**	**2.78 ± 1.71**	**2.03 ± 1.90**	**2.58 ± 2.20**
MA (11; 7)	3.75 ± 2.79	2.36 ± 1.52	2.70 ± 1.73	2.33 ± 2.16	2.76 ± 2.51
MA (11; 9)	3.95 ± 2.81	2.24 ± 1.59	2.78 ± 1.64	2.23 ± 2.05	3.00 ± 2.60
Loess (13)	4.66 ± 2.52	4.15 ± 2.30	2.85 ± 1.72	4.59 ± 2.87	6.25 ± 3.12
Loess (15)	4.03 ± 2.61	3.80 ± 2.01	2.80 ± 1.73	3.64 ± 2.64	4.84 ± 2.87
Loess (17)	3.46 ± 2.51	3.38 ± 2.04	2.76 ± 1.76	2.75 ± 2.12	3.58 ± 2.17
Loess (19)	3.20 ± 2.47	3.11 ± 1.90	2.79 ± 1.75	2.15 ± 1.90	2.87 ± 1.92
Loess (21)	**3.13 ± 2.50**	**2.76 ± 1.72**	**2.69 ± 1.78**	**2.00 ± 1.77**	**2.55 ± 2.03**
Loess (23)	3.22 ± 2.48	2.61 ± 1.64	2.71 ± 1.75	2.05 ± 1.81	2.52 ± 1.93
Proposed (13)	3.83 ± 2.36	2.65 ± 1.67	2.69 ± 1.77	3.01 ± 2.21	3.43 ± 2.13
Proposed (15)	3.56 ± 2.52	2.57 ± 1.57	2.63 ± 1.76	2.45 ± 2.08	2.90 ± 2.29
Proposed (17)	3.23 ± 2.51	2.55 ± 1.57	2.55 ± 1.74	2.00 ± 2.04	2.42 ± 2.01
Proposed (19)	**3.34 ± 2.63**	**2.57 ± 1.58**	**2.49 ± 1.70**	**1.99 ± 2.02**	**2.57 ± 2.03**
Proposed (21)	3.33 ± 2.48	2.45 ± 1.53	2.43 ± 1.69	2.05 ± 1.88	2.41 ± 2.04
Proposed (23)	3.38 ± 2.51	2.37 ± 1.57	2.44 ± 1.75	2.15 ± 1.95	2.52 ± 2.10

4.3.3 Feedback

One of the main goals for the proposed system was to provide feedback in real-time. Several aspects contribute to the total time needed to deliver information to the user. First of all, filtering introduces delay equal to half of filter length. The most efficient filtering was provided by filter lengths of 17 and 19, which corresponds to almost 300 ms. The average processing time for a single frame is less than 1 ms for both the Kinect and the IMU-based methods. The time required for the wireless transfer of the feedback information to the smartphone is negligible, as only the 7 values of the qualitative measurements are sent after analysis of the detected lunge. Therefore, the total time needed for providing feedback is approx. 300 ms, which is small enough to consider the system as working in real-time [204].

In order to measure the usefulness of the provided feedback, the mean and maximum values for the evaluated indirect parameters were calculated from both datasets, see Table 4.7. Since the Kinect provides data at 30 Hz, the duration of a single frame was approx. 33 milliseconds. Therefore, the values are given in both frames and milliseconds. The mean error for the hand timing is approx. 1.5 frames (50 ms) for the Kinect-based

method and approx. 2 frames (66 ms) for the IMU-based method, which is 6% and 8% of the maximum value for this parameter respectively. This indicates that both methods provide useful feedback in case of badly executed actions. Compared to the mean value, the errors constitute 34% and 45% respectively, which indicates that in a typical execution of an action, the feedback should be sufficient to determine if the arm was straightened before the forward motion, which was the main objective, although it is not accurate enough to allow for perfecting the hand timing. The mean error for duration is approx. 1.5 frames (50 ms) for the Kinect-based method and 2.5 frames (82 ms) for the IMU-based method, which is 7% and 12% of the mean value of this parameter respectively. This is sufficient for improving this parameter in the typical execution of the action, based on the provided feedback. The mean error for length (Kinect-based method only) is approx. 3.5 cm, which constitutes 4% of the mean value of this parameter, which allows for useful feedback, although, as stated before, this evaluation does not include errors introduced by the Kinect itself.

Table 4.7: Mean and maximum values for the indirect parameters, calculated jointly from both datasets.

	Mean	Maximum
Timing	4.44 fr. / 146 ms	24.00 fr. / 792 ms
Duration	20.21 fr. / 667 ms	34.00 fr. / 1122 ms
Length	83.82 cm	174.33 cm

Finally, the proposed system was evaluated by fencing coaches, who were satisfied with the provided feedback. The features of the system which were positively assessed include: real-time performance, qualitative measures for lunge actions, ability to get relevant feedback without a coach present. Another important aspect was that the system was competitive, therefore encouraging more practice in order to achieve better results. Expected improvements included wireless sensors and a smaller device for providing feedback (such as a smartwatch).

4.4 Summary

This chapter addressed the problem of the detection and analysis of actions in continuous, non-cyclic sport motion. Two dedicated datasets with continuous fencing footwork practice were recorded, the first one by using the Kinect and the second one by using both the Kinect and a custom system with two IMUs. Methods for detecting and analyzing lunge actions were developed, based on a velocity signal derived from the skeleton data provided by the Kinect, as well as the magnetic and inertial data provided by the IMUs. A novel model-based filter was proposed, which outperformed state-of-the-art filtering algorithms. The proposed system is able to accurately detect lunge actions and provides a number of related qualitative parameters, such as hand timing, lunge duration, lunge length, mean and maximum acceleration and speed. The signals are processed in real-time, therefore the feedback is instantaneous. Extensive experiments were conducted in order to verify the usability of the proposed solution. The evaluation was based on both a manually-prepared ground truth as well as the opinions of fencing coaches. The results indicate that the proposed system may be a valuable tool for use in training.

The initial results for the presented methods were presented during the 40th International Conference on Telecommunications and Signal Processing (TSP) held in Barcelona in July 2017 [175]. The publication of the proceedings resulted in the author being contacted by Dr Zoran Djurisic, main coach at Delta Fencing Center [67], Stockton, California, USA, who is conducting similar research by using inertial sensors to evaluate the fencers' performance. A collaboration with Delta Fencing Center was established, and new methods for motion analysis in fencing are being developed jointly.

Chapter 5

Immersive feedback for bladework practice in fencing using augmented reality

This chapter is devoted to providing real-time, immersive feedback for fencers practicing weapon techniques. A method for blade tracking in 3D is proposed, based on a single RGB camera and active markers. Model trajectories of weapon actions are recorded with fencing coaches and used for evaluation of fencing practice. Virtually generated trajectories are overlayed over the real-world view on augmented reality glasses. It is an innovative manner of providing real-time, immersive feedback, not used in sports before.

The problem of providing a real-time motion analysis and feedback in sports is rarely addressed in the literature. In [11] various types of visualisations were provided for rowing, table tennis and biathlon, based on the inertial and visual signals captured in real-time. The timing of the motion cycles in a gymnast training routine was measured and plotted in real-time in [226]. In regard to the fencing bladework, classification of the weapon actions was performed in [182] by employing a high-end motion-capture system, although no qualitative parameters were computed. The analysis of the upper limb biomechanics was conducted in [86] using both a motion-capture system and the EMG data. The authors focused on the arm-extension timing and its influence on the fencers' performance, therefore the weapon actions were not considered. The authors of [31] employ neural networks to distinguish between good and poor executions of the weapon actions, based on the inertial sensor readings, although no qualitative parameters are explicitly defined, which would be useful for correcting the actions during practice. There are no works in the literature regarding providing real-time, meaningful feedback, based on the qualitative analysis of the bladework.

The bladework analysis is a challenging problem due to several reasons. Firstly, the blade is difficult to track. It is thin, made of steel, which is a light-reflecting material, and moves very fast, which makes it hard to track with visual sensors. The high dynamics of the motion, including the frequent, rapid change of direction, result in noisy data when using IMUs. Moreover, it is difficult to capture the relevant qualitative parameters of the performed weapon actions. While the speed and range of the motion are important, the hand and weapon positioning during the action is crucial, but also more

difficult to evaluate. Providing real-time feedback in an efficient manner is another issue, since presenting the numerical parameters for the analyzed action may not be sufficient to support correcting the performed movement.

In this work, an efficient weapon tracking is realized with a single RGB camera and active markers. This approach enables reconstructing the trajectory of the blade during an action. The trajectories recorded with a coach are employed for building a model of an action, which is used for evaluation during practice. Visual cues and feedback are provided by presenting the expected and the actual trajectories of the performed actions on augmented reality (AR) glasses. The glasses are semi-transparent which allows mixing the real and virtual views, and therefore provide immersive feedback in an innovative manner [170]. The proposed methods for the blade tracking and learning action models were published in [171].

This chapter is organized as follows. Section 5.1 provides an overview of the proposed system. Section 5.2 presents the methods for the blade tracking, learning and evaluating weapon actions, and finally providing feedback with the AR glasses. The evaluation of the proposed methods is discussed in Section 5.3 and a summary is given in Section 5.4.

5.1 Overview

There are two types of the weapons used in fencing - thrusting and cutting (see Section 2.3.5). With the former only the thrusts score points, while with the latter both the cuts and thrusts can score points. The thrusting weapons require more precision, as even slight differences in the positioning or rotation of the blade can result in a weapon action being successful or not. For this reason, automatic training support is most beneficial for the thrusting weapons, and therefore this type of weapons is considered in this work.

There are several stages in practicing the weapon actions. At first, the novice fencers simply try to repeat the motion presented by a coach. Next, an action is practiced with a partner, who provides interaction, for instance by performing attacks in order for the first person to practice parries. Then, sequences of actions are performed, in which both persons are practicing, for example, attack - parry - counter-attack. Once the exercise is performed well-enough, the footwork is added, which increases difficulty due to the varying timing and distance. Finally, the actions are practiced during sparrings. The goal of the system presented in this work is to provide a support mainly for the initial stages of weapon action practice, as the process of reaching at least a medium level in weapon handling is very time-consuming and toilsome for the novice fencers.

Trajectories are an intuitive manner of understanding how a weapon action should be performed. When the novice fencers observe the weapon motion presented by a coach, they try to remember the trajectory of the tip of the blade. During practice they try to repeat it, although without constant supervision from the coach, who usually needs to share their attention between multiple students, they lack the feedback needed to correct their performance of the action. The goal of the system presented in this chapter is to provide them with a real-time trajectory-based feedback and enable efficient training in the absence of the coach.

Augmented reality enriches the perception of the real world with the virtually added

information. A popular approach to employing the AR is to display the virtual objects on the camera feed of the real-world, e.g. on a smartphone [301]. A significantly more immersive experience is provided by the AR glasses which are semi-transparent, and therefore allow to overlay the virtual objects not on the camera feed, but directly in the user's field of view. When looking trough the AR glasses, one can see the computer generated objects as if they were part of the surrounding environment. In this work, the AR glasses are employed to display the trajectories of the weapon actions in front of the practicing fencer. The coordinate mapping between the virtual and real world is performed, therefore during the weapon movement the trajectory is drawn along the tip of the blade, providing an accurate visualization of the motion. Mixing virtual and real environments in an interactive manner is sometimes referred to as mixed reality (MR) [80], although other authors consider AR to be a part of MR [267], therefore only the term augmented reality is used in this work.

Due to employing the AR glasses with the real-virtual coordinate mapping, it is possible to create the following practice routine. First, the coach performs multiple repetitions of an action which are used to build a model trajectory. Then, with the AR glasses, the model trajectory is displayed in front of the practicing fencer, who follows the displayed trajectory with the tip of the blade. The evaluation of the action is performed based on the difference between the model and current trajectories. In order to obtain the trajectories of the weapon actions, a reliable tracking of the blade is required. The AR glasses are equipped with a built-in camera, which provides a first-person perspective view. However, the quality of the video stream is not sufficient to employ object tracking algorithms based on the spatio-temporal interest points, such as SIFT [161] or SURF [14]. Therefore, LED markers are used, as they can be relatively easily detected.

The architecture of the proposed system is based on employing the following devices: a double-LED marker, placed near the tip of the blade, the AR glasses for the practicing fencer, and a laptop, connected wirelessly to the glasses (see Fig. 5.1). The camera in the AR glasses is used for detecting the marker in real-time, which enables tracking the trajectories of the practiced weapon actions. The laptop provides a graphical user control interface (GUI) during the calibration and model recording procedures. Employing a laptop facilitated experiments with the prototype system implemented in this work, however, it would be possible for the system to operate without a laptop - this would require implementing the control interface on the glasses themselves or on a mobile device. An additional benefit of using a laptop is that is allows to preview the tracked trajectories on the screen, and therefore provides feedback for a coach as well.

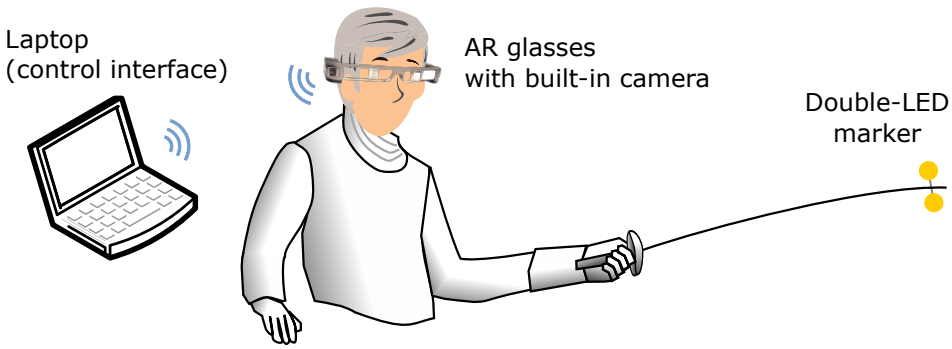

Fig. 5.1: Architecture of the proposed system (graphics from: [85, 290]).

5.2 Methods

This section discusses methods used for the blade tracking, learning models of the weapon actions, evaluating bladework practice and providing immersive feedback with the augmented reality.

5.2.1 Blade tracking

Based on consultations with fencing experts, there are three parameters that describe the motion during the weapon actions: the trajectory of the tip of the blade, the trajectory of the base of the blade, and the rotation of the blade. The first one is most relevant for the offensive actions, in terms of evading the opponent's blade and hitting the intended target area. The second and the third are most relevant for the defensive actions, where the proper positioning and rotation of the blade is required for the parry actions to be effective. All three parameters are important during action sequences, when the fencers quickly and constantly change between the offensive and defensive actions, or even perform offensive-defensive actions.

In this work, two of these parameters are tracked during fencers' practice, due to the limitation of the employed devices. Although the camera built in the AR glasses has wide angle lens, the view it provides is not sufficient for the reliable tracking of the base of the blade, as during weapon practice, the base of the blade is often outside the camera's view area. Therefore, only the tip of the blade and the rotation are tracked. While this is a considerable limitation, the visual cues and feedback provided by the proposed system are still useful for the bladework training support. Also, this limitation results strictly from capabilities of the used device and not from the proposed method itself. Having a camera with a wider angle, it would be possible to implement tracking of an additional marker and visualization of another trajectory.

The relatively low quality of the video stream provided by the built-in camera makes the tracking of the weapon blade difficult. The fast motion results in blurred images, particularly in poor lighting conditions (which are typical for training halls), due to the relatively long exposure times required by the light-sensitive matrix to capture a video frame. For this reason, the algorithms which are based on detecting the spatio-temporal interest points, such as SIFT [161] or SURF [14], are not applicable in this case. Instead, LED markers are mounted on the blade and the camera is set to low exposition, which results in the captured image being dark, except for the light sources. Therefore, the LED markers are easily detectable by finding the brightest pixels in the image, as long as there are no other light sources in the camera's view. Additionally, due to the low exposition setting, the exposure times are shorter and consequently the images are less blurred.

Since only a single point is tracked - the tip of the blade - a single LED marker is sufficient to find this point in an image. However, it would only provide 2D tracking. Instead, a double-LED marker is used, which enables the depth and rotation tracking as well. The marker, mounted on the tip of the blade, has two LEDs, approx. 5 cm from each other, which are placed perpendicularly to the blade (see Fig. 5.2). Based on the relative position of the LEDs in the captured image it is possible to estimate both the depth and rotation, as will be explained further in this section. The drawback of this approach is that not all weapon actions can be practiced, i.e. those requiring contact

between the tip of the blade and the partner's blade cannot be performed. In exercises without a partner this limitation does not apply. The marker itself is very simple and low cost, as it contains two LEDs soldered to a CR2032 battery case (see Fig. 5.2).

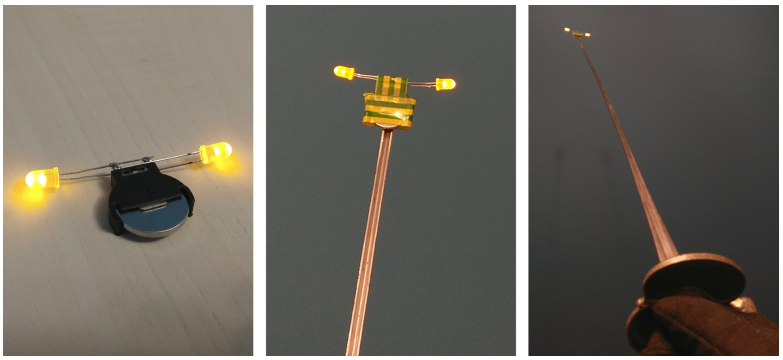

Fig. 5.2: Double-LED marker (left) mounted on the tip of the blade (middle) and the entire weapon view (right).

The detection of the LEDs in the captured images is performed in two steps. Firstly, the image is converted to grayscale, and a threshold operation is applied, that creates a binary image, in which all the pixels above the threshold are white and all other are black (see Fig. 5.3). The threshold is selected automatically during a short calibration procedure. The user puts the marker in the camera view and starts the calibration from the control interface. An image is captured and then the lowest and highest thresholds are found, for which the number of the detected LEDs is correct. The final threshold is set as the lowest threshold plus 1/10 of the difference between the highest and the lowest threshold. This allows to avoid other objects in the image being above the threshold, but also to capture the marker during fast motion, when the LEDs are blurred in the image and therefore appear to be less bright. The computations for this procedure last less than 1 second. Alternatively, the threshold can be set manually, in a dedicated display mode, where all the pixels which are above the threshold are shown.

The second step of the LEDs detection consists in a clustering. Once the binary image is created, clusters of white pixels are found, which correspond to the markers. The algorithm runs through all pixels in the image and for each found white pixel it marks it as a new cluster and performs the region growing operation [2] using this pixel as a seed point. All white neighboring pixels are added to the cluster, changed to black, and then the algorithm is run recursively on their 8 neighbors. This methods assumes, that all pixels in a cluster are connected and the separate clusters are not connected. Based on the conducted experiments, in the considered scenario this is almost always the case. The other cases are handled as follows. When more than two clusters are found, only the largest two are considered, therefore discarding the additional small clusters corresponding to separate pixels, which are not connected to the main two clusters. When less than one cluster is found, the frame is discarded.

The proposed tracking method is based solely on detection, therefore spatio-temporal dependencies between consecutive frames are not considered. As discussed in Section 5.3.1 this approach is sufficient, when no other light sources are present in the camera's field of view. However, in order to handle other light sources and reflections, such dependencies may used, e.g. by employing particle filters [144].

The centers of the detected clusters are referred to as keypoints, each corresponding to a single LED (see Fig. 5.3). In the current setup, two keypoints corresponding to the double-LED marker are detected, although in general, more keypoints could be found with the proposed method. The 2D position of the tip of the blade is calculated as the middle point between the two keypoints. The depth and rotation are calculated based on the relative positions of the keypoints. The closer the tip of the blade to the camera is, the greater the pixel distance between the keypoints (see Fig. 5.4). By performing a one-time calibration it is possible to map the pixel distance between the keypoints to the real distance between the marker and the camera. In a limited depth range the relation is approximately proportional, therefore it is sufficient to record several images with known marker-camera distances and measure the keypoints' relative pixel distance in each image, in order to find the parameters for a linear equation describing their dependency. The rotation is given by the angle between the line connecting the two keypoints and the horizontal axis of the camera (see Fig. 5.4). Due to the view angle of the camera this is not exactly the same as the rotation measured relatively to the ground. However, since both the model learning and the practice evaluation depend on the rotation measured with this method, the perceived rotation is more relevant than the rotation relative to the ground.

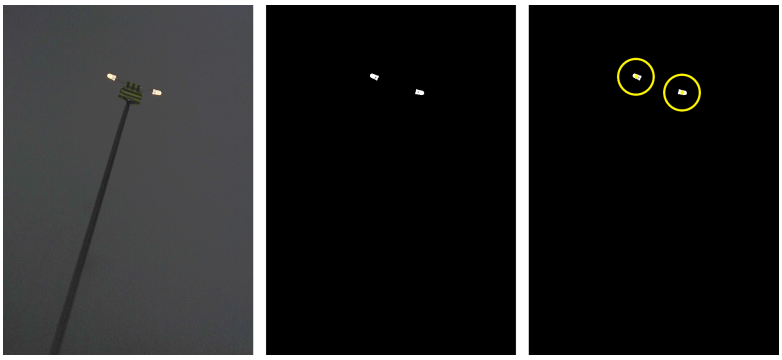

Fig. 5.3: Marker detection. The camera is set to the low exposition setting (left), the thresholding operation is applied (middle) and the LED positions (keypoints) are detected after clustering (right).

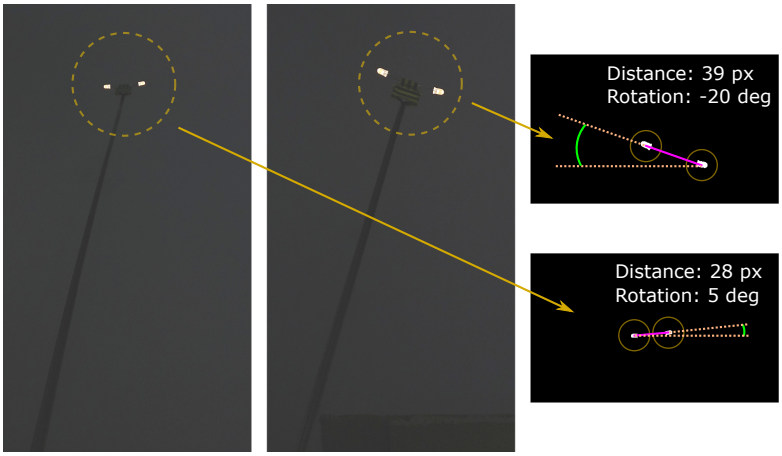

Fig. 5.4: Depth and rotation estimation based on the detection of the relative positions of the two LEDs. The pixel distance between the LEDs corresponds to the depth - smaller distance (left) indicates that the tip of the blade is further away.

It is worth noting, that other methods for determining the depth and rotation were considered as well. The recent models of AR glasses include depth sensors, which would be an obvious choice for the depth estimation. However, the initial experiments with the Kinect showed, that the detection of the thin, light-reflective blade was very poor, as in most frames it was not visible on the depth map. The rotation could be tracked by attaching an inertial sensor to the base of the weapon, although this would made the system both more complex to use and more expensive. Also, inertial sensors do not provide distance information, therefore employing IMU would not be sufficient to eliminate the need for a double-LED marker.

5.2.2 Action models

Due to the fast motion of the blade and the camera acquisition rate equal to 30 Hz, the detection of the double-LED marker provides only sparse sampling of the blade tip's trajectory (see Fig. 5.5 left). The distances between the consecutive points depend on the changes in the speed of the performed action and the overall length of the trajectory depends on the time of execution. For both the visualization and evaluation purposes, dense trajectories with a common length are required. Therefore, the cubic spline interpolation [65] is applied, which provides densely sampled trajectories, with a constant, arbitrary chosen number of points (see Fig. 5.5 right).

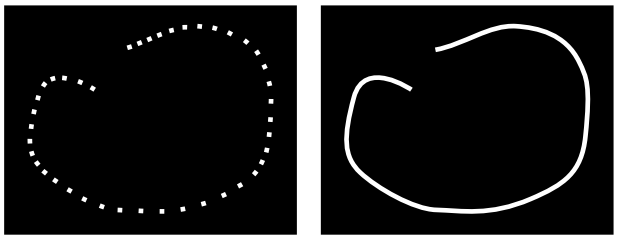

Fig. 5.5: Tracking of the tip of the blade: the detected marker positions (left) and the interpolated trajectory (right).

The use of the interpolated trajectories allows to calculate a model trajectory for an action, and employ a point-by-point comparison with the trajectories recorded during bladework practice. However, a common start point is required as well. For this reason, building a model of an action starts with defining the start point. The fencer simply puts the tip of the blade in the desired position and saves the start point using the control interface. Each repetition of the action, during both the model learning and practice evaluation, begins with moving the tip of the blade to the start point displayed on the AR glasses.

In order to facilitate using the system, the action repetitions are detected automatically. A finite state machine (FSM) is employed (see Fig. 5.6). The initial state is *detection*, in which the tip of the blade is detected, but the trajectories are not recorded. Once the tip of the blade is moved to the displayed start point, the state changes to *in position*, and a timer is started which after 1 second changes the state to *ready*. Then, the system waits for the blade movement to start, which triggers the transition to the *recording* state. Each state is indicated to the user by changing the color of the virtual start point marker.

The *in position* and *ready* states create a time buffer between moving the tip of the blade to the start point and the beginning of the action, which makes the system more convenient to use. In the *recording* state, in each frame, the position of the detected tip

of the blade is added to the list of the tracked points. While the action is performed, a simplified trajectory is displayed on the AR glasses, by connecting the detected points with straight lines. The action recording finishes when the fencer stops moving the blade. Then, the interpolation is performed and the interpolated trajectory is displayed instead of the simplified one. The next repetition of the action begins when the tip of the blade is moved again to the start point.

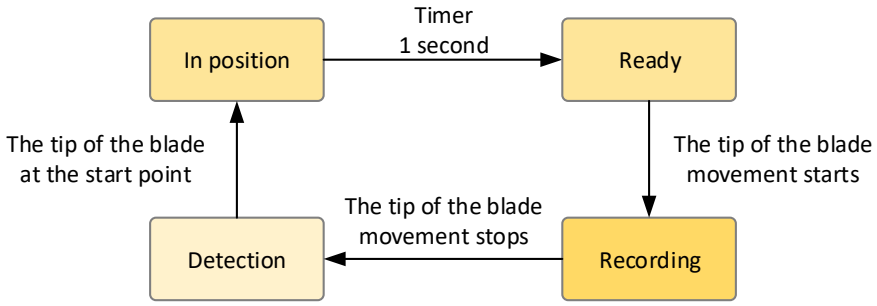

Fig. 5.6: Finite state machine used to automatically detect action repetitions.

Detection of the blade movement is performed as follows. The tip of the blade is considered to start moving, when the average displacement of the detected points in the last 20 frames is above a pre-defined threshold. The threshold for detecting the movement was chosen experimentally, taking into account that even when the weapon is not moved voluntarily, the tip of the blade is not perfectly still. Two users were asked to hold the weapon in place for 10 seconds and then to perform small movements for another 10 seconds. The displacement of the detected marker corresponded, equally, to both intentional and unintentional movements, therefore average displacement was used as the threshold. The stop of the movement is detected when the average displacement in the last 20 frames is below the threshold.

There are two modes of using the proposed system, model learning and practice evaluation. The first one is dedicated to building action models based on the input provided by the fencing coaches. The coach selects a start point and then performs multiple repetitions of the chosen action. The mean trajectory is calculated point-wise, based on all repetitions. Given: N - number of the recorded trajectories (action repetitions), L - length of an interpolated trajectory, $p_{i,k}$ - *i-th* point of the *k-th* trajectory, points m_i of the mean trajectory are computed as follows:

$$\forall i \in 1...L, m_i = \frac{\sum_{k=1}^{N} p_{i,k}}{N} \tag{5.1}$$

The standard deviations s_i are computed in each point as well, based on the Euclidean distance between the points in the mean trajectory and each recorded repetition of the action:

$$\forall i \in 1...L, s_i = \sqrt{\frac{\sum_{k=1}^{N} (p_{i,k} - m_i)^2}{N}} \tag{5.2}$$

In the practice evaluation mode, the recorded model of an action is loaded, the mean trajectory is displayed, and the fencer tries to repeat it. Both the current and the model trajectories are visible, therefore an instant visual feedback is provided. Once a repetition of the practiced action is finished, the trajectories are compared numerically. For each point of the current interpolated trajectory, its distance to the corresponding

point in the model trajectory is calculated. The accuracy in each point is proportional to the ratio between the calculated distance to the model point and the standard deviation for this point, which is also stored in the model. The edge values are a 100% when the distance is lower than the standard deviation, and a 0% when the distance is higher than twice the standard deviation. Given: c_i - *i-th* point in the current trajectory, the accuracy a_i of this point is computed as follows:

$$\forall i \in 1...L, a_i = 100 - ((min(max(\frac{m_i - c_i}{s_i}, 1), 2) - 1) * 100) \tag{5.3}$$

The accuracy for the entire trajectory is computed as the mean from all points. Both 2D and 3D distances were considered for the point-wise trajectory comparison. Since the depth in the AR glasses was not perceived by the users well enough, 2D Euclidean distances in *xy* plane were eventually employed. The rotation is evaluated in a similar manner. The mean value and standard deviation for the rotation angle is stored for each point of the model trajectory and point-by-point comparison is performed in the practice mode. Separate values for the trajectory and the rotation accuracies are displayed. For the purpose of visualization the rotation is also color-coded.

5.2.3 Augmented reality

In the proposed system, visual cues and feedback are provided to the practicing fencers by employing AR glasses. The Epson Moverio BT-300 device was used in this work [78]. It includes light-weight AR glasses connected with a wire to a small, Android-based processing unit, which is also used for control (see Fig. 5.7). The glasses have a semi-transparent display for each eye - the user can see both the surrounding environment and the displayed content. The black color in the generated image is transparent on the glasses, therefore seamless mixing of the real and virtual views is possible. The glasses operate with 1280 x 720 resolution, 30 Hz refresh rate and provide the field of view of 23 degrees. A 5 megapixel camera is built in the glasses, on the right side. The processing unit has 1.44 GHz quad core Intel Atom processor, 2 GB RAM and 16 GB of internal memory. It is also equipped with a touchpad and several control keys, acting as the user interface. Additionally, both the glasses and the processing unit have built-in inertial sensors, although these were not used in this work.

Fig. 5.7: Epson Moverio BT-300 AR device includes control and processing unit (left) and semi-transparent AR glasses (right).

The glasses can operate in two modes - either by displaying the same or different images for each eye. In the first case, users see a flat screen floating in front of them. The second mode has the capability of stimulating 3D perception of the generated objects. Humans perceive depth that is determined on the basis of stereo vision. Since each eye looks from a slightly different point, they receive slightly different images. The depth of the seen objects is estimated based on the disparity in the left and right images (see Fig. 5.8). This mechanism can be used to generate 3D virtual scenes, by displaying for each eye an image of the scene generated from a shifted viewpoint [55]. In the proposed system, this was implemented with the OpenGL ES [209], which is a dedicated 3D library for

the Android system. In the OpenGL ES the scene view is generated by specifying the position and direction of a virtual camera. Therefore, it is sufficient to slightly reposition the virtual camera when generating the images for the left and the right eye in order to provide depth perception for the generated objects.

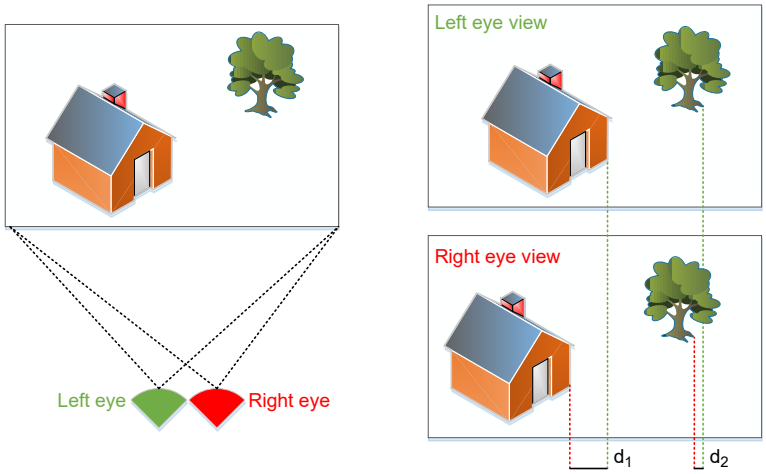

Fig. 5.8: Depth perception in stereo vision. The image disparity between the left and right eye is greater for the house (d_1), than for the tree (d_2), which indicates that the house is closer than the tree.

The main difficulty in creating a mixed real-virtual view with the AR glasses is to generate the virtual objects in the proper positions relatively to the real environment. A simple case would be to generate an object, that would appear to float over a flat surface. In order to do that, a translation between the real-world and virtual-world coordinate systems must be provided. In the proposed system, the virtually generated trajectories are supposed to follow the double-LED marker placed on the tip of the blade. Therefore, the system needs to calculate the 3D position of each point in the virtual trajectory, based on the marker position provided by the camera, in such a manner, that they would both appear to the fencer to be in the same location in the mixed real-virtual view (see Fig. 5.9). In order to make this possible, a dedicated calibration procedure is proposed.

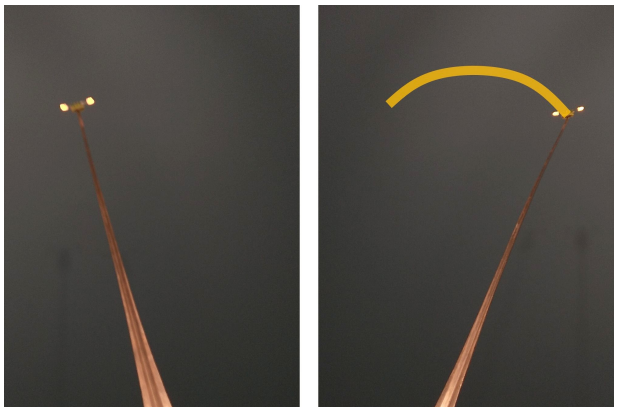

Fig. 5.9: Expected mixed real-virtual view. The virtually generated trajectory should match the motion of the tip of the blade in the real-world view.

The 3D position of the marker (corresponding to the tip of the blade) seen by the camera is defined by x and y coordinates of the middle point between the two keypoints

detected in the image, as well as the pixel-distance between them, which corresponds to the depth (see Section 5.2.1). The virtual object coordinates are given to the OpenGL ES visualization engine in an arbitrary defined coordinate system. Based on the position of the virtual camera the engine calculates the projection of the object in the images displayed for the left and the right eye. In order to create the mapping between the two coordinate systems, the user manually matches the real and the virtual objects in several points in space during the calibration procedure. The collected data is used to calculate the coordinate mapping parameters. For each calibration point, an object (a small triangle) is displayed on the AR glasses and the user is asked to move the tip of the blade to this position, and then click a button in the control interface, that saves the calibration data for this point, which include both virtual and real coordinates.

In order to calculate all the required parameters, ten points in space are used in the calibration process. In perspective vision, the field of view for close objects is smaller than the field of view for distant objects, e.g. a hand held close to the face occupies more of the field of view than a human seen in a distance. Similarly, translation of the coordinates from the real to virtual world in the xy plane depends on the z distance of the real object. Therefore, the calibration starts with the depth. The virtual marker is displayed in the middle of the screen, in a close distance, for the first calibration point, and in a far distance for the second calibration point. These distances correspond roughly to the typical distance of the tip of the blade before and after the arm extension, which is approx. 130 cm and 170 cm respectively (although exact values depend on the fencer). Two calibration points allow to find the parameters for the linear equation describing the relation between the virtual and real depth coordinates. Given z_{rc} - depth in the real-world coordinate system for the close distance, z_{rf} - depth in the real-world coordinate system for the far distance, z_{vc} - depth in the virtual-world coordinate system for the close distance, z_{vf} - depth in the virtual-world coordinate system for the far distance, the parameters a_z and b_z are computed from the pair of equations:

$$z_{vc} = a_z * z_{rc} + b_z \tag{5.4}$$

$$z_{vf} = a_z * z_{rf} + b_z \tag{5.5}$$

The general equation for computing the virtual depth z_v based on the real depth z_r is as follows:

$$z_v = a_z * z_r + b_z \tag{5.6}$$

Next, four calibration points are gathered for the close distance and another four for the far distance. The four points are in the middle of the left, right, top and bottom edges of the displayed area (see Fig. 5.10). Therefore, four sets of parameters for linear equations are calculated corresponding to the horizontal and the vertical coordinates in the close and the far distances: (a_{xc}, b_{xc}), (a_{yc}, b_{yc}), (a_{xf}, b_{xf}), (a_{yf}, b_{yf}). Given x_{rc}, y_{rc}, x_{rf}, y_{rf} - x and y real-world coordinates in the close and far distances, x_{vc}, y_{vc}, x_{vf}, y_{vf} - x and y virtual-world coordinates in the close and far distances, the resulting equations are as follows:

$$x_{vc} = a_{xc} * x_{rc} + b_{xc} \tag{5.7}$$

$$y_{vc} = a_{yc} * y_{rc} + b_{yc} \tag{5.8}$$

$$x_{vf} = a_{xf} * x_{rf} + b_{xf} \tag{5.9}$$

$$y_{vf} = a_{yf} * y_{rf} + b_{yf} \tag{5.10}$$

Equations 5.7, 5.8, 5.9, 5.10 allow to compute x and y virtual coordinates in the close or far distance only (see Fig. 5.10). General equations for computing x_v and y_v coordinates at any distance can be expressed as follows:

$$x_v = a_x * x_r + b_x \tag{5.11}$$

$$y_v = a_y * y_r + b_y \tag{5.12}$$

where:

$$w = (z_v - z_{vc})/(z_{vf} - z_{vc}) \tag{5.13}$$

$$a_x = a_{xc} * (w - 1) + a_{xf} * w \tag{5.14}$$

$$a_y = a_{yc} * (w - 1) + a_{yf} * w \tag{5.15}$$

$$b_x = b_{xc} * (w - 1) + b_{xf} * w \tag{5.16}$$

$$b_y = b_{yc} * (w - 1) + b_{yf} * w \tag{5.17}$$

Once the calibration procedure is finished, the coordinate mapping can be performed as follows. Firstly, the virtual depth is calculated with the depth equation (Eq. 5.6). Then, based on the depth weight (Eq. 5.13), the parameters for the horizontal and vertical equations are determined (Eq. 5.14, 5.15, 5.16, 5.17). Finally, the virtual x and y coordinates are computed (Eq. 5.11 and 5.12).

With the obtained coordination mapping, calculated during the calibration procedure, it is possible to display virtually generated trajectory which follows the tip of the blade. During training, the fencers see the model trajectory of the practiced action, which constitutes a visual cue to how to perform it correctly, as well as the current trajectory of their execution of the action, which provides instant, immersive feedback. Once the fencer finishes the action, the system compares the current trajectory with the model (see Section 5.2.2) and displays a numerical evaluation score, for both the trajectory and rotation. The rotation is additionally color-coded: 45 degrees to the right is pure green, 45 degrees to the left is pure blue, and the colors are mixed in between. Edge rotation values were chosen based on the recommendation from the fencing coaches, who stated that in the majority of actions the blade's rotation should be between these values.

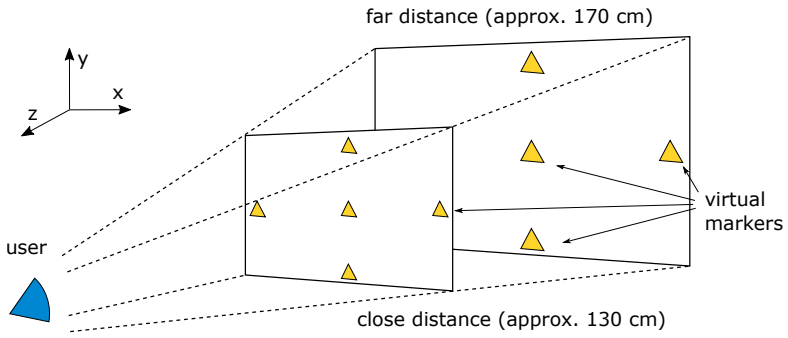

Fig. 5.10: Calibration points are gathered in two distances (close and far) by displaying virtual markers in 10 different positions, which the user has to match with the physical weapon.

5.3 Experiments and results

In this section the proposed methods for the blade tracking, model learning, practice evaluation and providing feedback are verified in experiments. The results are discussed, including the opinions from fencing coaches, fencers and non-fencers who helped to evaluate the proposed system.

5.3.1 Blade tracking

In order to verify the proposed blade tracking method, short recordings of bladework were acquired at three different locations: a university laboratory and two different training halls. The recordings, each lasting approx. 10 seconds, included typical weapon actions performed with a weapon equipped with the double-LED marker, and were captured with the camera built in the AR glasses, with the resolution set to 640 x 480. At each location, a place for the experiments was chosen, such that there would be no other light sources or reflections present in the camera's view. In all video frames the positions of the LEDs were manually labeled, in order to provide the ground truth. The threshold for the automatic detection was set automatically with the proposed calibration procedure (see Section 5.2.1). The position of the tip of the blade, the pixel distance between the LEDs, and the rotation were computed for both manually labeled and automatically detected LED positions. The average differences between the automatic detection and the ground truth, including the standard deviation, are presented in Table 5.1. The detection rate is presented as well, computed as the percentage of the frames in which both LEDs were detected correctly. A LED is considered to be detected correctly when the difference between the automatically found position and the ground truth is smaller than half of the distance between the two LEDs in the ground truth.

Table 5.1: Blade tracking results computed for recordings from 3 different locations. The detection rate indicates the percentage of the frames for which both LEDs were correctly found. For the tip position, the distance between the LEDs and the rotation, average differences between the automatic detection and the ground truth are given, including the standard deviation.

Parameter	Location 1 (University lab.)	Location 2 (Training hall A)	Location 3 (Training hall B)	Average
Nb. of frames	307	214	294	272
Detection [%]	100	100	97.96	99.32
Tip [px]	1.07 ± 0.29	1.22 ± 0.45	1.21 ± 0.51	1.17 ± 0.42
Distance [px]	0.45 ± 0.27	0.62 ± 0.31	0.85 ± 2.39	0.64 ± 0.99
Rotation [deg]	0.84 ± 0.76	1.03 ± 0.93	0.90 ± 1.19	0.92 ± 0.96

The average error in finding the tip of the blade was slightly above 1 pixel, in all recordings. The distance between the LEDs was estimated with the average error a little above half pixel. For reference, the image size was 640 x 480 and the pixel distance between the LEDs was typically between 20 to 30 pixels. The average rotation value error was smaller than 1 degree. The results indicate, that the proposed detection method provides high accuracy. In regard to the detection rate, there were only a few frames,

in which the LEDs were not found correctly. The detailed analysis revealed, that in those frames the motion was very fast, resulting in a very blurred images and therefore the LEDs appearing much darker. Typically, during practice, weapon actions are not performed with maximum speed, therefore this issue occurs rarely.

Other possible issues were identified as well. Firstly, when the weapon is moving very fast, causing blurred images, and the rotation of the blade matches its direction, both LEDs may form a single cluster of white pixels in the image. Such cases could be handled by analyzing the shape of the cluster, although they occur very rarely, therefore this was not addressed. Secondly, reflections on the blade may occur and could be recognized as other LEDs. In the conducted experiments it was sufficient to choose a place away from the light sources, so that the reflections did not occur, although it may not always be possible. In most weapon actions such reflections can be handled by simply ignoring all the white-pixel clusters found in the image, except for the two nearest to the top edge of the image, as the tip of the blade is almost always closer to the top edge of the image, than the rest of the blade. Using very bright LEDs could also allow to distinguish between the LEDs and the reflections, based on the brightness difference, although too bright LEDs can be inconvenient for the practicing fencer. Another simple solution would be to cover the blade with a non-reflective tape. A relevant issue are also additional light sources or reflective objects. None can be present in the field of view of the camera, which may by a limitation when using the system during trainings.

The proposed method for the depth estimation, which is based on the pixel distance between the LEDs in the captured image, was verified in the following manner. The AR glasses were set to look at a measuring tape placed on the floor and the weapon with the double-LED marker was moved along the tape, with 10 cm intervals (see Fig. 5.11). The pixel distance between the LEDs in the captured images was computed automatically using the proposed detection method. Figure 5.12 presents the LEDs' relative pixel distance plotted against their actual distance from the camera. In the measured range, which was 50 to 170 cm, the dependency is non-linear, due to the wide angle lens in the AR glasses' camera (see Fig. 5.12 left). However, during bladework practice the tip of the blade is typically between 130 and 170 cm, and for such smaller range, the dependency can be approximated with a linear function (see Fig. 5.12 right), which indicates, that the proposed depth estimation method is appropriate for the considered scenario.

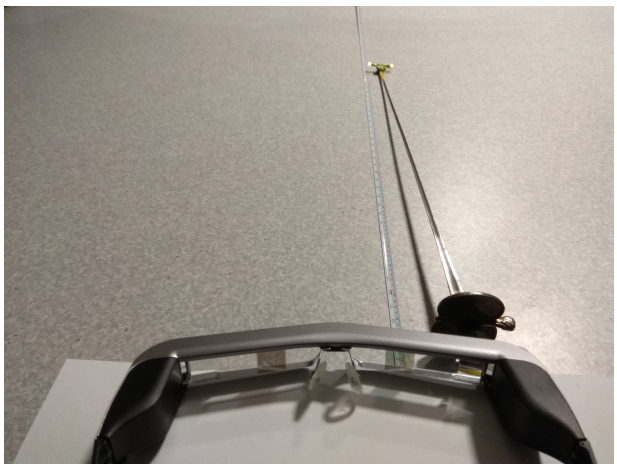

Fig. 5.11: Setup for the depth estimation experiments.

The proof-of-concept implementation of the proposed system is able to operate in real-time. With the camera resolution set to 640 x 480 the average processing time for a single frame is 36 ms, which provides an almost smooth operation. With the camera resolution set to 320 x 240 the average processing time for a single frame is 13 ms, which results in a completely smooth operation. It is worth noting, the even with the lower resolution, the user perception of the accuracy of the blade tracking is similar as with the higher resolution, therefore the lower resolution is sufficient.

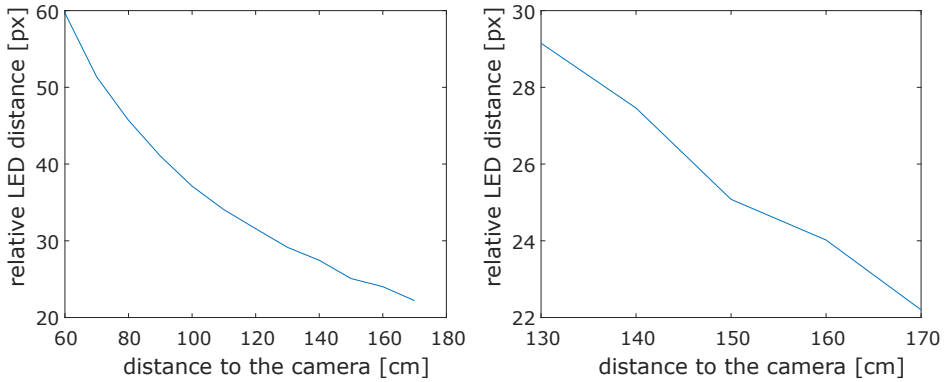

Fig. 5.12: Relative pixel distance of the detected LEDs plotted against their actual distance from the camera. The full measured range (left) and the range typical for bladework practice (right).

5.3.2 Action models

The purpose of the action models is to provide visual cues, as well as to allow for the numerical evaluation of the bladework practice. The goal of the numerical evaluation is to provide high scores when the performed trajectory is similar to the model trajectory, and low scores otherwise. The ground truth for the similarity is difficult to define. In the bladework practice it comes down to human perception. Since the system is supposed to provide similar feedback as a coach, it should consider the trajectories to be similar in those cases, in which a coach would consider them to be. Therefore, in order to verify the proposed method for the action evaluation, a dedicated tool was implemented, which allows to simulate actions by drawing with a mouse on a computer screen. The sixth-to-forth parry action was chosen for the evaluation, as it is one of the most commonly used weapon actions and it can be performed without forward motion, therefore the trajectory can be evaluated in 2D. The model for the action was created based on 20 repetitions of drawing with the mouse. The mean trajectory is presented in Fig. 5.13 in red, with the standard deviation indicated by the circles drawn around selected points. Several correct and incorrect actions were drawn and compared to the model trajectory, as presented in Fig. 5.13. Based on this experiment, it was concluded, that the automatic estimation of the trajectory similarity corresponds to the human assessment. It is worth noting, that the proposed method considers both shape and range of the trajectory, as both are important in bladework practice.

The FSM for the automatic detection of the start and stop of an action repetition was evaluated based on the users opinions. Although the users required a few action repetitions to get used to it, the final conclusion was that it is rather convenient and easy to use, and allows to practice without any additional control interface. Based on the experiments with the fencers, one other feature was added to the system, namely model

editor, which allows to choose which of the recorded repetitions should be included in the final model. This was required, because sometimes an incorrect repetition of an action was performed during the model learning.

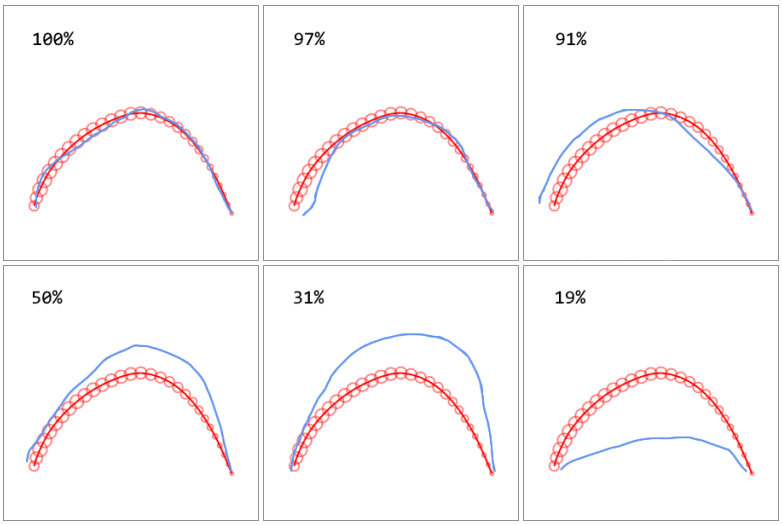

Fig. 5.13: Verification of the automatic evaluation of the similarity of the trajectories. The red lines indicate the model trajectory, and the red circles indicate the standard deviation. The blue lines indicate the trajectories of the practiced actions. The average percentage similarity is provided.

5.3.3 Augmented reality

The purpose of employing the augmented reality was to provide an immersive feedback for the practicing fencers, by creating a mixed virtual-real view, with the virtually generated trajectories properly aligned with the real-world weapon. The mixed view, captured with a camera looking trough the AR glasses is presented in Fig. 5.14. The evaluation presented in this section is based on user opinions, regarding the perception of the mixed view, as well as the usefulness of the system for supporting bladework practice. Three fencing coaches, four advanced fencers and four people, who had no prior experience with fencing, participated in the experiments. The advanced fencers and the non-fencers participated as the test subjects in evaluating the bladework practice mode of the system. The coaches provided input for recording the action models and evaluated the performance of the persons practicing weapon actions with the support of the proposed system.

Fig. 5.14: Actual mixed real-virtual view displayed on the AR glasses. The virtually generated trajectory is overlayed on the real-world view.

The calibration procedure for the coordinate mapping was performed with three persons. Each of these persons was then asked to assess the accuracy of the blade tracking using each of the three calibrated coordinate mappings. All three persons chose the same coordinate mapping as the most accurate. This indicates, that the calibration process is not user-specific. By repeating the calibration procedure several times, it was revealed that the most important factor in the calibration is the precise matching of the tip of the blade with the virtual marker. Therefore, it may be beneficial to gather more calibration points, in order to minimize the influence of the matching errors. A manual fine-tuning of the coordinate mappings could be useful as well.

The selected coordinate mapping was employed in the subsequent experiments. All participants were asked to assess the accuracy of the tracking of the tip of the blade, first in the static positions and then during the movement. A virtual marker, namely a small triangle, was displayed in the estimated position of the tip of the blade. For the horizontal direction in the static positions the virtual triangle was always present between the LEDs, mostly in the middle, and sometimes closer to one of the LEDs. In the vertical direction, the accuracy of the tracking was similar. The depth was estimated less accurately, as the object appeared sometimes slightly too close or too far. Also, the depth tracking was less stable, as the virtual marker slightly oscillated, even when the blade was not moving. This was due to the slight differences in the keypoint positions estimation between frames, which result from the noise in the camera video stream. In a test recording lasting 30 seconds, in which the blade was stationary, the variance of the estimated keypoints distance was approx. 1 pixel, which corresponds to 5 cm.

In regard to the trajectories drawn during the motion, all users stated, that they correspond very well to the performed motion. The accuracy of the depth estimation was not an issue in this case, as small differences in the depth are unnoticeable in the generated trajectories. The system provided smooth operation, although a time delay was occurring between the movement of the real weapon and the following virtual object (the marker or the trajectory, depending on the mode of operation). The marker, or the last point of the trajectory, were displayed in the position, in which the tip of the blade was a moment earlier. Detailed profiling revealed, that this was caused by the camera, which delivers the images with delay.

The next step of the evaluation of the proposed system was to verify, if the visual clues and feedback result in performing the bladework practice more correctly. Three weapon action models were recorded with the fencing coaches: 6th-to-4th parry 5.15, 4th-to-6th parry, 6th-to-4th parry with a riposte. The first two included a semi-circular, horizontal motion and a small forward motion, while the last one included additional significant forward motion. The initial experiments showed, that the human depth perception of the displayed trajectories is very limited, as they are flat and therefore provide little 3D context. The stereo vision itself is not sufficient to provide relevant depth perception in this case. For this reason, the trajectories were compared only in 2D and, separately, with regard to the rotation. Therefore, the last action, 6th-to-4th parry with riposte, was not used in the experiments. It was suggested by one of the coaches, that the depth could be color-coded, similarly as the rotation, only with different colors, e.g. red and yellow. This will be investigated in the future work.

The advanced fencers were asked to practice the two selected weapon actions with the support of the proposed system. A few minutes were needed for each person, to get used

to the system. After several repetitions, the fencers were able to repeat the trajectories and rotations, with relatively high average score, ranging from 70% to 90%. This indicates, that the system is able to evaluate the fencer's performance relatively well. Then, the system was introduced to the non-fencers and they were asked to perform weapon practice as well. The average score varied significantly between the persons, ranging from 20% to 90%. Since the non-fencers did not have any fencing skills, their scores corresponded to their general manual skills, which explains the varied results.

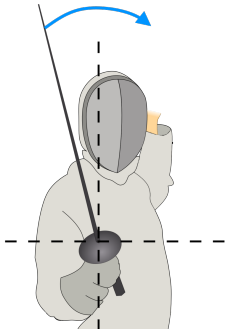

Fig. 5.15: 6th-to-4th parry action in fencing (graphics from: [292]).

The weapon practice performed by the non-fencers was also evaluated by the coaches. The most important observation was that all persons performed the actions more correctly than typical novice fencers. The novice fencers tend to perform the parry motion too widely and with significantly too little or too much rotation. The visual cues to the rotation and the movement range, provided by the proposed system, largely prevent such errors. However, the motion of the base of the blade required corrections from the coach. As discussed in Section 5.2.1, the motion of the base of the blade is one of the key factors in the weapon action performance, however it is not considered by the proposed system, due to the hardware limitations.

In regard to the general assessment of the proposed system, a few issues were indicated by the participants of the experiments. First of all, the field of view in which the virtual objects can be displayed is very limited and therefore does not allow to perform the bladework actions in a full range. This problem can be addressed only by employing different AR glasses, with a larger display. It is worth noting, that this is an expected direction of the development for such devices. It was also pointed out, that with the double-LED marker on the tip of the blade it is not possible to practice some of the weapon actions with a partner. This problem could also be addressed by employing a better AR device, capable of capturing the video stream with a wider angle, which would allow to mount the two LEDs on the weapon's guard and only a single LED on the tip of the blade. A potential issue was the lack of compensation for the head movement, which can introduce apparent motion of the blade. However, no head movement is typically present during bladework practice, and no such problems occurred.

Two additional benefits of using the proposed system were observed in the experiments. Firstly, some of the participants were greatly motivated by the displayed evaluation score, which indicates that the system encourages perfecting the weapon actions. Secondly, the trajectory visualization on the laptop proved to be interesting for the coaches, who stated that recording the weapon actions with the fencers would make it possible to evaluate and compare their techniques.

5.4 Summary

This chapter was devoted to providing real-time, immersive feedback in fencing bladework practice. In order to obtain this goal, several novel methods were introduced, in regard to the tracking of the 3D position of the tip of the blade, as well as the blade rotation, learning weapon action models, evaluating bladework performance based on the learned models, and finally providing fencers with a mixed virtual-real view, with the use of the AR glasses. The proposed methods were implemented with a proof-of-concept system, which operates in real-time. The system was evaluated in several experiments, which included participation of fencing coaches, fencers, and non-fencers. The proposed methods proved to be suitable for aiding the bladework practice. It is worth noting, that employing the AR glasses for providing immersive feedback is an innovative approach for supporting sport training.

Chapter 6

Conclusions

This chapter concludes the dissertation by providing a summary of the conducted research, the verification of the thesis statement, and a discussion of advantages and limitations of the proposed methods. Additionally, possible directions for future work are presented.

As a result of this dissertation, novel methods were introduced for the automatic analysis of sports motion, which allow for feedback useful for improving sports skills. Three relevant research subjects were investigated: recognition of similar sports actions based on dynamics; temporal segmentation and qualitative analysis of actions in continuous sports practice; and providing immersive feedback by employing augmented reality and action model learning on the basis of object tracking. All proposed algorithms were evaluated in an extensive set of experiments. The proposed methods are intended for a single sports discipline, namely fencing, although they can be adapted or can constitute a starting point for developing methods for handling similar issues in other disciplines as well.

6.1 Thesis statement verification

The thesis of the dissertation, formulated in Chapter 1, states that automatic analysis of techniques and body motion patterns in sports can result in feedback, which would allow for improving sports skills. An extensive survey of state-of-the-art motion and sports analysis methods revealed a number of challenges in this area, which so far have not been properly tackled. The methods devised in this work, targeted specifically at sports actions, address these challenges in order to verify the proposed thesis statement.

The first challenge was the recognition of sports-specific motion. This issue is investigated in Chapter 3, which concerns fencing lunge action classification. It is shown, that by using the proposed methods, which are based on dynamics analysis, it is possible to effectively distinguish between different types of a lunge, even though the motion is very similar. The ability to automatically classify similar sports actions could be used in training for the practicing person to learn how to correctly perform all variations of an action, as well as in tactical analysis, with respect to the choice of an action variation in different situations.

The second challenge concerned the temporal segmentation of the body motion patterns in sports and qualitative analysis of sports actions. This is addressed in Chapter 4, by proposing novel methods for the detection and analysis of a lunge action in a continuous fencing footwork training routine. As verified by the conducted experiments, the proposed algorithms allow for effective detection of the lunge action segments in both depth and inertial data, as well as provide a number of relevant qualitative parameters. Moreover, the feedback is delivered to the athlete in real-time, and can therefore be used for improving sports skills continuously during training.

The final challenge was providing immersive, real-time feedback which would constitute an innovative manner of supporting sports training. In Chapter 5, augmented reality glasses are employed for creating a mixed real-virtual view, in which visual cues for the correct weapon action execution are presented in the form of virtually generated trajectories. With the proposed methods for blade tracking, model learning, and real-virtual coordinate mapping, all operating in real-time, it is possible to accurately integrate the virtual objects and the real-world environment in the mixed view in the augmented reality glasses. Therefore, immersive feedback is provided, which supports bladework practice in a very intuitive manner.

The results presented in the thesis provide evidence that the goal of the dissertation was achieved, i.e. it was confirmed that automatic analysis of motion in sports can result in feedback relevant for improving sports skills.

6.2 Contributions and limitations

The main contribution of this thesis is the introduction of methods for sports motion analysis, which allow for providing useful feedback during sports practice. The proposed methods address several different issues.

The recognition of sports-specific motion is addressed in Chapter 3. While the classification of general actions usually concerns significantly different actions, in sports, even minor differences in motion can constitute the basis for distinguishing between the performed techniques. Therefore, new recognition methods based on action dynamics were proposed. Several novel feature extraction algorithms were introduced, for both visual and inertial data. Additionally, new approaches to feature selection and fusion were presented. In an extensive evaluation, based on a dedicated fencing footwork dataset, it was shown that the proposed methods provide efficient recognition of sports actions, despite state-of-the-art general action classification methods failing to do so. The proposed methods provided also superior results on a publicly available UTD-MHAD dataset. An additional contribution is the fencing footwork dataset, recorded specifically for this work, and made publicly available for other researchers [166]. The proposed methods were described in three papers [174, 176, 177].

Another discussed issue was temporal segmentation. State-of-the-art motion analysis methods consider, almost exclusively, pre-segmented actions, while the process of segmentation itself is rarely addressed. The methods proposed in Chapter 4 provide highly efficient detection of relevant actions in continuous fencing footwork practice. In the same chapter, a qualitative analysis of the actions is performed. While general motion analysis focuses on the recognition of actions, in sports measuring the motion execution performance is crucial. In this work, the parameters relevant for the detected fencing

footwork actions are identified and measured with high accuracy. Real-time feedback is delivered wirelessly to a smartphone, making it available during the training. The proposed methods were published in [175].

Finally, an innovative manner of providing real-time, immersive feedback is introduced for supporting bladework practice. The proposed methods allow to accurately track the blade in 3D with a single RGB camera in order to record weapon action models and evaluate weapon practice by comparing it with the models. The feedback is provided by creating a mixed real-virtual view on the AR glasses. By employing the proposed calibration procedure for the real-virtual coordinate mapping, virtually generated trajectories are aligned with the weapon seen in the actual environment. To the best of the author's knowledge, this is the first attempt at employing AR with mixed, real-virtual view for sports training support. The proposed methods for the blade tracking and learning action models were published in [171].

The proposed methods are not without limitations. The recognition accuracy for similar actions indicates that there is still room for improvement. The temporal segmentation is currently limited to detecting a single action. The augmented reality-based system would benefit from having a greater field of view and a better camera, which could result in the system being useful in a wider range of weapon action exercises. The proof-of-concept implementations require further work in order to make them more convenient for use in day-to-day training sessions. Nevertheless, the current methods provide meaningful feedback for the fencers, which is a considerable achievement.

6.3 Future work

There are several possible directions of conducting further research related to the discussed subject. First of all, the prototype of the system for the detection and analysis of fencing footwork actions, presented in Chapter 4, can be developed further, in order to make it more convenient to use during training and possibly also in competitions. This would not only provide a useful tool for fencing footwork training, but also allow to easily gather more data which could be used for further development of action analysis methods. The detection of other actions and the analysis of their performance would be interesting. It is worth noting, that joint research in this area is currently being conducted with the Delta Fencing Center, located in California, USA [67].

Alternative means of providing feedback are also worth consideration. As stated in Chapter 4, smartwatches have great potential in this matter, since they are small, light, have built-in inertial sensors, provide a good quality display, allow for wireless connection, and are easily available. Also, they are becoming more and more popular. Another interesting manner of providing feedback would be virtual reality (VR). By tracking the athletes' motion, virtual exercises could be possible, maybe even including virtual opponents, controlled by artificial intelligence algorithms. The visual cues and feedback for bladework practice could be presented by using VR as well. The main advantage in this case would be a much lower cost - while AR glasses are expensive, VR can be achieved with a simple low-cost cardboard adapter used with a smartphone [93].

Finally, it would be beneficial to adapt the results of this research to other sports disciplines. Since similar problems occur in other sports as well, it should be possible to develop dedicated motion analysis methods based on the ones proposed in this work.

Bibliography

[1] Abdi H., Williams L.J.: Principal component analysis. In: *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2(4), pp. 433–459, 2010.

[2] Adams R., Bischof L.: Seeded region growing. In: *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 16(6), pp. 641–647, 1994.

[3] Agarwal A., Triggs B.: Recovering 3D human pose from monocular images. In: *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 28(1), pp. 44–58, 2006.

[4] Ahmadi A., Mitchell E., Richter C., Destelle F., Gowing M., O'Connor N.E., Moran K.: Toward automatic activity classification and movement assessment during a sports training session. In: *IEEE Internet of Things J.*, vol. 2(1), pp. 23–32, 2015.

[5] Akula A., Shah A.K., Ghosh R.: Deep learning approach for human action recognition in infrared images. In: *Cognitive Systems Research*, vol. 50, pp. 146–154, 2018.

[6] Altun K., Barshan B., Tunçel O.: Comparative study on classifying human activities with miniature inertial and magnetic sensors. In: *Pattern Recognition*, vol. 43(10), pp. 3605–3620, 2010.

[7] Amaro J.P., Patrão S.: A survey of sensor fusion algorithms for sport and health monitoring applications. In: *42nd Annual Conf. of the IEEE Industrial Electronics Society (IECON)*, pp. 5171–5176. IEEE, 2016.

[8] Annadani Y., Rakshith D., Biswas S.: Sliding dictionary based sparse representation for action recognition. In: *arXiv preprint arXiv:1611.00218*, 2016.

[9] Aramis Fencing School. `https://aramis.pl/`. Last access on Jan 2019.

[10] Avci A., Bosch S., Marin-Perianu M., Marin-Perianu R., Havinga P.: Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey. In: *23rd Int. Conf. on Architecture of Computing Systems (ARCS)*, pp. 1–10. VDE, 2010.

[11] Baca A., Kornfeind P.: Rapid feedback systems for elite sports training. In: *IEEE Pervasive Computing*, vol. 5(4), pp. 70–76, 2006.

[12] Baccouche M., Mamalet F., Wolf C., Garcia C., Baskurt A.: Sequential deep learning for human action recognition. In: *Int. Workshop on Human Behavior Understanding*, pp. 29–39. Springer, 2011.

[13] Bao L., Intille S.: Activity recognition from user-annotated acceleration data. In: *Pervasive Computing*, pp. 1–17, 2004.

[14] Bay H., Ess A., Tuytelaars T., Van Gool L.: Speeded-up robust features (SURF). In: *Computer Vision and Image Understanding*, vol. 110(3), pp. 346–359, 2008.

[15] Beauchemin S.S., Barron J.L.: The computation of optical flow. In: *ACM Computing Surveys (CSUR)*, vol. 27(3), pp. 433–466, 1995.

[16] Belongie S., Malik J., Puzicha J.: Shape matching and object recognition using shape contexts. In: *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24(4), pp. 509–522, 2002.

[17] Bengio Y., Simard P., Frasconi P.: Learning long-term dependencies with gradient descent is difficult. In: *IEEE Trans. on Neural Networks*, vol. 5(2), pp. 157–166, 1994.

[18] Berndt D.J., Clifford J.: Using dynamic time warping to find patterns in time series. In: *3rd Int. Conf. on Knowledge Discovery and Data Mining (KDD)*, vol. 10(16), pp. 359–370, 1994.

[19] Blank M., Gorelick L., Shechtman E., Irani M., Basri R.: Actions as space-time shapes. In: *10th IEEE Int. Conf. on Computer Vision (ICCV)*, vol. 2, pp. 1395–1402. IEEE, 2005.

[20] Blanz V., Vetter T.: Face recognition based on fitting a 3D morphable model. In: *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25(9), pp. 1063–1074, 2003.

[21] Bloom V., Makris D., Argyriou V.: G3D: A gaming action dataset and real time action recognition evaluation framework. In: *IEEE Conf. on Computer Vision and Pattern Recognition Workshops (CVPR Workshops)*, pp. 7–12. IEEE, 2012.

[22] Bober T., Rutkowska-Kucharska A., Jaroszczuk S., Barabasz M., Woźnica W.: Kinematic characterisation of the lunge and the fleche in epee fencing: two case studies. In: *Polish J. of Sport and Tourism*, vol. 23(4), pp. 181–185, 2016.

[23] Bobick A.F., Davis J.W.: The recognition of human movement using temporal templates. In: *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 23(3), pp. 257–267, 2001.

[24] Borges P.V., Conci N., Cavallaro A.: Video-based human behavior understanding: A survey. In: *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 23(11), pp. 1993–2008, 2013.

[25] Bortz J.E.: A new mathematical formulation for strapdown inertial navigation. In: *IEEE Trans. on Aerospace and Electronic Systems*, vol. AES-7(1), pp. 61–66, 1971.

[26] Borysiuk Z., Piechota K., Minkiewicz T.: Analysis of performance of the fencing lunge with regard to the difficulty level of a technical-tactical task. In: *J. of Combat Sports and Martial Arts*, vol. 4(2), pp. 135–139, 2013.

[27] Bottoms L., Greenhalgh A., Sinclair J.: Kinematic determinants of weapon velocity during the fencing lunge in experienced épée fencers. In: *Acta of Iboengineering and Biomechanics*, vol. 15(4), pp. 109–113, 2013.

[28] Boyle M., Edwards C., Greenberg S.: The effects of filtered video on awareness and privacy. In: *ACM Conf. on Computer Supported Cooperative Work*, pp. 1–10. ACM, 2000.

[29] Bulbul M.F., Jiang Y., Ma J.: DMMs-based multiple features fusion for human action recognition. In: *Int. J. of Multimedia Data Engineering and Management*, vol. 6(4), pp. 23–39, 2015.

[30] Cahill-Rowley K., Rose J.: Temporal-spatial reach parameters derived from inertial sensors: Comparison to 3D marker-based motion capture. In: *J. of Biomechanics*, vol. 52, pp. 11–16, 2017.

[31] Campaniço A.T., Valente A., Serôdio R., Escalera S.: Data's hidden data: qualitative revelations of sports efficiency analysis brought by neural network performance metrics. In: *Motricidade*, vol. 14(4), pp. 94–102, 2018.

[32] Chan J.S., Wong A.C., Liu Y., Yu J., Yan J.H.: Fencing expertise and physical fitness enhance action inhibition. In: *Psychology of Sport and Exercise*, vol. 12(5), pp. 509–514, 2011.

[33] Chandrashekar G., Sahin F.: A survey on feature selection methods. In: *Computers & Electrical Engineering*, vol. 40(1), pp. 16–28, 2014.

[34] Chaquet J.M., Carmona E.J., Fernández-Caballero A.: A survey of video datasets for human action and activity recognition. In: *Computer Vision and Image Understanding*, vol. 117(6), pp. 633–659, 2013.

[35] Chatfield K., Simonyan K., Vedaldi A., Zisserman A.: Return of the devil in the details: Delving deep into convolutional nets. In: *arXiv preprint arXiv:1405.3531*, 2014.

[36] Chaudhry R., Ravichandran A., Hager G., Vidal R.: Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In: *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 1932–1939. IEEE, 2009.

[37] Chen C., Jafari R., Kehtarnavaz N.: Improving human action recognition using fusion of depth camera and inertial sensors. In: *IEEE Trans. on Human-Machine Systems*, vol. 45(1), pp. 51–61, 2015.

[38] Chen C., Jafari R., Kehtarnavaz N.: UTD-MHAD: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor. In: *IEEE Int. Conf. on Image Processing (ICIP)*, pp. 168–172. IEEE, 2015.

[39] Chen C., Jafari R., Kehtarnavaz N.: A real-time human action recognition system using depth and inertial sensor fusion. In: *IEEE Sensors J.*, vol. 16(3), pp. 773–781, 2016.

[40] Chen C., Jafari R., Kehtarnavaz N.: A survey of depth and inertial sensor fusion for human action recognition. In: *Multimedia Tools and Applications*, vol. 76(3), pp. 4405–4425, 2017.

[41] Chen H.S., Chen H.T., Chen Y.W., Lee S.Y.: Human action recognition using star skeleton. In: *4th ACM Int. Workshop on Video Surveillance and Sensor Networks*, pp. 171–178. ACM, 2006.

[42] Chen H.T., He Y.Z., Chou C.L., Lee S.Y., Lin B.S., Yu J.Y.: Computer-assisted self-training system for sports exercise using Kinects. In: *IEEE Int. Conf. on Multimedia and Expo Workshops (ICMEW)*, pp. 1–4. IEEE, 2013.

[43] Chen H.T., Tsai W.J., Lee S.Y., Yu J.Y.: Ball tracking and 3D trajectory approximation with applications to tactics analysis from single-camera volleyball sequences. In: *Multimedia Tools and Applications*, vol. 60(3), pp. 641–667, 2012.

[44] Chen J., Little J.J.: Where should cameras look at soccer games: Improving smoothness using the overlapped hidden Markov model. In: *Computer Vision and Image Understanding*, vol. 159, pp. 59–73, 2017.

[45] Chen L., Wei H., Ferryman J.: A survey of human motion analysis using depth imagery. In: *Pattern Recognition Letters*, vol. 34(15), pp. 1995–2006, 2013.

[46] Chen Y.P., Yang J.Y., Liou S.N., Lee G.Y., Wang J.S.: Online classifier construction algorithm for human activity detection using a tri-axial accelerometer. In: *Applied Mathematics and Computation*, vol. 205(2), pp. 849–860, 2008.

[47] Chen Z., Zhu Q., Soh Y.C., Zhang L.: Robust human activity recognition using smartphone sensors via CT-PCA and online SVM. In: *IEEE Trans. on Industrial Informatics*, vol. 13(6), pp. 3070–3080, 2017.

[48] Cheng X., Zhuang X., Wang Y., Honda M., Ikenaga T.: Particle filter with ball size adaptive tracking window and ball feature likelihood model for ball's 3D position tracking in volleyball analysis. In: *Pacific Rim Conf. on Multimedia*, pp. 203–211. Springer, 2015.

[49] Cleveland W.S.: Robust locally weighted regression and smoothing scatterplots. In: *J. of the American Statistical Association*, vol. 74(368), pp. 829–836, 1979.

[50] Cloete T., Scheffer C.: Benchmarking of a full-body inertial motion capture system for clinical gait analysis. In: *30th Annual Int. Conf. of the IEEE Engineering in Medicine and Biology Society (EMBS)*, pp. 4579–4582. IEEE, 2008.

[51] Coalter F.: *A Wider Social Role for Sport: Who's Keeping the Score?* Routledge, 2007.

[52] Cortes C., Vapnik V.: Support-vector networks. In: *Machine Learning*, vol. 20(3), pp. 273–297, 1995.

[53] Counting your steps. `https://www.10000steps.org.au/articles/counting-steps`. Last access on Jan 2019.

[54] Cuesta-Vargas A.I., Galán-Mercant A., Williams J.M.: The use of inertial sensors system for human motion analysis. In: *Physical Therapy Reviews*, vol. 15(6), pp. 462–473, 2010.

[55] Cyganek B., Siebert J.P.: *An introduction to 3D computer vision techniques and algorithms.* John Wiley & Sons, 2011.

[56] Czajkowski Z.: *Szermierka na Szpady: Technika, Taktyka, Trening, Walka.* Wydawnictwo Sport i Turystyka, 1977.

[57] Czajkowski Z.: *Theory, Practice and Methodology in Fencing: Advanced Course for Fencing Coaches.* Wydawnictwo AWF Katowice, 2001.

[58] Czajkowski Z.: *Nauczanie Techniki Sportowej.* Centralny Ośrodek Sportu, 2004.

[59] Czajkowski Z.: *Understanding Fencing. The Unity of Theory and Practice.* SKA Swordplay Books, 2005.

[60] Dalal N., Triggs B.: Histograms of oriented gradients for human detection. In: *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 886–893. IEEE, 2005.

[61] Dalal N., Triggs B., Schmid C.: Human detection using oriented histograms of flow and appearance. In: *European Conf. on Computer Vision (ECCV)*, pp. 428–441. Springer, 2006.

[62] Danafar S., Gheissari N.: Action recognition for surveillance applications using optic flow and SVM. In: *Asian Conf. on Computer Vision*, pp. 457–466. Springer, 2007.

[63] Davis J.W., Bobick A.F.: The representation and recognition of human movement using temporal templates. In: *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 928–934. IEEE, 1997.

[64] Dawar N., Kehtarnavaz N.: Action detection and recognition in continuous action streams by deep learning-based sensing fusion. In: *IEEE Sensors J.*, vol. 18(23), pp. 9660–9668, 2018.

[65] De Boor C.: *A Practical Guide to Splines*, vol. 27. Springer-Verlag New York, 1978.

[66] Dejnabadi H., Jolles B.M., Aminian K.: A new approach to accurate measurement of uniaxial joint angles based on a combination of accelerometers and gyroscopes. In: *IEEE Trans. on Biomedical Engineering*, vol. 52(8), pp. 1478–1484, 2005.

[67] Delta Fencing Center. `http://www.deltafencingcenter.com/`. Last access on Jan 2019.

[68] Di Russo F., Taddei F., Apnile T., Spinelli D.: Neural correlates of fast stimulus discrimination and response selection in top-level fencers. In: *Neuroscience Letters*, vol. 408(2), pp. 113–118, 2006.

[69] Dollár P., Rabaud V., Cottrell G., Belongie S.: Behavior recognition via sparse spatio-temporal features. In: *2nd Joint IEEE Int. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pp. 65–72. IEEE, 2005.

[70] Drory A., Zhu G., Li H., Hartley R.: Automated detection and tracking of slalom paddlers from broadcast image sequences using cascade classifiers and discriminative correlation filters. In: *Computer Vision and Image Understanding*, 2016.

[71] E-gym applications. `https://egym.com/en/apps/`. Last access on Jan 2019.

[72] Eckardt F., Münz A., Witte K.: Application of a full body inertial measurement system in dressage riding. In: *J. of Equine Veterinary Science*, vol. 34(11), pp. 1294–1299, 2014.

[73] Efros A.A., Berg A.C., Mori G., Malik J.: Recognizing action at a distance. In: *IEEE Int. Conf. on Computer Vision (ICCV)*, p. 726. IEEE, 2003.

[74] El Madany N.E.D., He Y., Guan L.: Human action recognition via multiview discriminative analysis of canonical correlations. In: *IEEE Int. Conf. on Image Processing (ICIP)*, pp. 4170–4174. IEEE, 2016.

[75] Ellis C., Masood S.Z., Tappen M.F., LaViola J.J., Sukthankar R.: Exploring the trade-off between accuracy and observational latency in action recognition. In: *Int. J. of Computer Vision*, vol. 101(3), pp. 420–436, 2013.

[76] Elmadany N.E., He Y., Guan L.: Human gesture recognition via bag of angles for 3D virtual city planning in CAVE environment. In: *18th IEEE Int. Workshop on Multimedia Signal Processing (MMSP)*, pp. 1–5. IEEE, 2016.

[77] Enzweiler M., Gavrila D.M.: Monocular pedestrian detection: Survey and experiments. In: *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31(12), pp. 2179–2195, 2009.

[78] Epson BT-300 smart glasses. `https://www.epson.eu/products/see-through-mobile-viewer/moverio-bt-300?productfinder=bt300`. Last access on Jan 2019.

[79] Escobedo E., Camara G.: A new approach for dynamic gesture recognition using skeleton trajectory representation and histograms of cumulative magnitudes. In: *29th Conf. on Graphics, Patterns and Images (SIBGRAPI)*, pp. 209–216. IEEE, 2016.

[80] Farshid M., Paschen J., Eriksson T., Kietzmann J.: Go boldly!: Explore augmented reality (AR), virtual reality (VR), and mixed reality (MR) for business. In: *Business Horizons*, vol. 61(5), pp. 657–663, 2018.

[81] Feichtenhofer C., Pinz A., Zisserman A.: Convolutional two-stream network fusion for video action recognition. In: *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 1933–1941. 2016.

[82] Figat J., Kornuta T., Kasprzak W.: Performance evaluation of binary descriptors of local features. In: *Int. Conf. on Computer Vision and Graphics*, pp. 187–194. Springer, 2014.

[83] FIVB: Challenge system regulations for volleyball. `https://ebook.cev.eu/development/Referee/CEV_Challenge%20System_Regulations_Volleyball.pdf`. Last access on Jan 2019.

[84] Fox K.R.: The influence of physical activity on mental well-being. In: *Public Health Nutrition*, vol. 2(3a), pp. 411–418, 1999.

[85] Freepik vectorpocket. `https://www.freepik.com/vectorpocket`. Last access on Jan 2019.

[86] Frère J., Göpfert B., Nüesch C., Huber C., Fischer M., Wirz D., Friederich N.: Kinematical and EMG-classifications of a fencing attack. In: *Int. J. of Sports Medicine*, vol. 32(01), pp. 28–34, 2011.

[87] Ganapathi V., Plagemann C., Koller D., Thrun S.: Real time motion capture using a single time-of-flight camera. In: *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 755–762. IEEE, 2010.

[88] Gerke S., Linnemann A., Müller K.: Soccer player recognition using spatial constellation features and jersey number recognition. In: *Computer Vision and Image Understanding*, 2017.

[89] Geronimo D., Lopez A.M., Sappa A.D., Graf T.: Survey of pedestrian detection for advanced driver assistance systems. In: *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 32(7), pp. 1239–1258, 2010.

[90] Gers F.A., Schraudolph N.N., Schmidhuber J.: Learning precise timing with LSTM recurrent networks. In: *J. of Machine Learning Research*, vol. 3(Aug), pp. 115–143, 2002.

[91] Ghasemzadeh H., Loseu V., Guenterberg E., Jafari R.: Sport training using body sensor networks: A statistical approach to measure wrist rotation for golf swing. In: *4th Int. Conf. on Body Area Networks*, p. 2. ICST Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, 2009.

[92] Gholipour M., Tabrizi A., Farahmand F.: Kinematics analysis of lunge fencing using stereophotogrametry. In: *World J. of Sport Sciences*, vol. 1(1), pp. 32–37, 2008.

[93] Google: Cardboard. `https://vr.google.com/cardboard/`. Last access on Jan 2019.

[94] Gorelick L., Blank M., Shechtman E., Irani M., Basri R.: Actions as space-time shapes. In: *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29(12), pp. 2247–2253, 2007.

[95] Guo Y., Tao D., Liu W., Cheng J.: Multiview cauchy estimator feature embedding for depth and inertial sensor-based human action recognition. In: *IEEE Trans. on Systems, Man, and Cybernetics: Systems*, vol. 47(4), pp. 617–627, 2017.

[96] Guo Y., Xu G., Tsuji S.: Understanding human motion patterns. In: *12th IAPR Int. Conf. on Pattern Recognition, Conf. B: Computer Vision and Image Processing*, vol. 2, pp. 325–329 vol.2, 1994.

[97] Guyon I., Elisseeff A.: An introduction to variable and feature selection. In: *J. of Machine Learning Research*, vol. 3(Mar), pp. 1157–1182, 2003.

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

[99] Hahm G.J., Cho K.: Event-based sport video segmentation using multimodal analysis. In: *Int. Conf. on Information and Communication Technology Convergence (ICTC)*, pp. 1119–1121. IEEE, 2016.

[100] Hämäläinen P.: Interactive video mirrors for sports training. In: *3rd Nordic Conf. on Human-Computer Interaction*, pp. 199–202. ACM, 2004.

[101] Hamatani T., Sakaguchi Y., Uchiyama A., Higashino T.: Player identification by motion features in sport videos using wearable sensors. In: *9th Int. Conf. on Mobile Computing and Ubiquitous Networking (ICMU)*, pp. 1–6. IEEE, 2016.

[102] Hanai Y., Nishimura J., Kuroda T.: Haar-like filtering for human activity recognition using 3D accelerometer. In: *13th IEEE Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop (DSP/SPE)*, pp. 675–678. IEEE, 2009.

[103] Hardegger M., Ledergerber B., Mutter S., Vogt C., Seiter J., Calatroni A., Tröster G.: Sensor technology for ice hockey and skating. In: *IEEE 12th Int. Conf. on Wearable and Implantable Body Sensor Networks (BSN)*, pp. 1–6. IEEE, 2015.

[104] Harris C., Stephens M.: A combined corner and edge detector. In: *4th Alvey Vision Conf.*, vol. 15, pp. 10–5244. Manchester, UK, 1988.

[105] Hassan M.M., Uddin M.Z., Mohamed A., Almogren A.: A robust human activity recognition system using smartphone sensors and deep learning. In: *Future Generation Computer Systems*, vol. 81, pp. 307–313, 2018.

[106] HawkEye system. `https://www.hawkeyeinnovations.com`. Last access on Jan 2019.

[107] Hedayati M., Cree M.J., Scott J.: Scene structure analysis for sprint sports. In: *Int. Conf. on Image and Vision Computing New Zealand (IVCNZ)*, pp. 1–5. IEEE, 2016.

[108] Henry P., Krainin M., Herbst E., Ren X., Fox D.: RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments. In: *The Int. J. of Robotics Research*, vol. 31(5), pp. 647–663, 2012.

[109] Herath S., Harandi M., Porikli F.: Going deeper into action recognition: A survey. In: *Image and Vision Computing*, vol. 60, pp. 4–21, 2017.

[110] Herpin G., Gauchard G.C., Lion A., Collet P., Keller D., Perrin P.P.: Sensorimotor specificities in balance control of expert fencers and pistol shooters. In: *J. of Electromyography and Kinesiology*, vol. 20(1), pp. 162–169, 2010.

[111] Hibbs A., O'Donoghue P.: Strategy and tactics in sports performance. In: *Routledge handbook of sports performance analysis*, pp. 266–276. Routledge, 2013.

[112] Hinton G.E., Osindero S., Teh Y.W.: A fast learning algorithm for deep belief nets. In: *Neural Computation*, vol. 18(7), pp. 1527–1554, 2006.

[113] Ho T.K.: Random decision forests. In: *3rd Int. Conf. on Document Analysis and Recognition*, vol. 1, pp. 278–282. IEEE, 1995.

[114] Hopfield J.J.: Neural networks and physical systems with emergent collective computational abilities. In: *Proceeding of the National Academy of Science of the United States of America*, vol. 79(8), pp. 2554–2558, 1982.

[115] Horan S.A., Evans K., Morris N.R., Kavanagh J.J.: Thorax and pelvis kinematics during the downswing of male and female skilled golfers. In: *J. of Biomechanics*, vol. 43(8), pp. 1456–1462, 2010.

[116] Hosoe H., Sako S., Kwolek B.: Recognition of JSL finger spelling using convolutional neural networks. In: *15th IAPR Int. Conf. on Machine Vision Applications (MVA)*, pp. 85–88. IEEE, 2017.

[117] HowStuffWorks: How fencing equipment works. `https://entertainment.howstuffworks.com/fencing-equipment3.htm`. Last access on Jan 2019.

[118] HowStuffWorks: How olympic timing works. `https://entertainment.howstuffworks.com/olympic-timing1.htm`. Last access on Jan 2019.

[119] Hu M.K.: Visual pattern recognition by moment invariants. In: *IEEE Trans. on Information Theory*, vol. 8(2), pp. 179–187, 1962.

[120] I2C Bus. `https://www.i2c-bus.org/`. Last access on Jan 2019.

[121] Jain A., Zongker D.: Feature selection: Evaluation, application, and small sample performance. In: *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19(2), pp. 153–158, 1997.

[122] Jarvie G.: *Sport, culture and society: An introduction.* Routledge, 2013.

[123] Ji S., Xu W., Yang M., Yu K.: 3D convolutional neural networks for human action recognition. In: *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 35(1), pp. 221–231, 2013.

[124] Jian M., Zhang S., Wu L., Zhang S., Wang X., He Y.: Deep key frame extraction for sport training. In: *Neurocomputing*, vol. 328, pp. 147–156, 2019.

[125] Jiang Y.G., Dai Q., Xue X., Liu W., Ngo C.W.: Trajectory-based modeling of human actions with motion reference points. In: *European Conf. on Computer Vision (ECCV)*, pp. 425–438, 2012.

[126] Johansson G.: Visual perception of biological motion and a model for its analysis. In: *Perception & Psychophysics*, vol. 14(2), pp. 201–211, 1973.

[127] Kapela R., Świetlicka A., Rybarczyk A., Kolanowski K., et al.: Real-time event classification in field sport videos. In: *Signal Processing: Image Communication*, vol. 35, pp. 35–45, 2015.

[128] Karpathy A., Toderici G., Shetty S., Leung T., Sukthankar R., Fei-Fei L.: Large-scale video classification with convolutional neural networks. In: *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 1725–1732. 2014.

[129] Kasiri S., Fookes C., Sridharan S., Morgan S.: Fine-grained action recognition of boxing punches from depth imagery. In: *Computer Vision and Image Understanding*, 2017.

[130] Kasprzak W., Wilkowski A., Czapnik K.: Hand gesture recognition based on free-form contours and probabilistic inference. In: *Int. J. of Applied Mathematics and Computer Science*, vol. 22(2), pp. 437–448, 2012.

[131] Ke Q., An S., Bennamoun M., Sohel F., Boussaid F.: SkeletonNet: Mining deep part features for 3-D action recognition. In: *IEEE Signal Processing Letters*, vol. 24(6), pp. 731–735, 2017.

[132] Keerthi S., Shevade S., Bhattacharyya C., Murthy K.: Improvements to Platt's SMO algorithm for SVM classifier design. In: *Neural Computation*, vol. 13(3), pp. 637–649, 2001.

[133] Kellokumpu V., Zhao G., Pietikäinen M.: Human activity recognition using a dynamic texture based method. In: *British Machine Vision Conf. (BMVC)*, vol. 1, p. 2. 2008.

[134] Khaire P., Kumar P., Imran J.: Combining CNN streams of RGB-D and skeletal data for human activity recognition. In: *Pattern Recognition Letters*, 2018.

[135] Kim H.J., Lee J.S., Yang H.S.: Human action recognition using a modified convolutional neural network. In: *Int. Symposium on Neural Networks*, pp. 715–723. Springer, 2007.

[136] Kim Y., Cho K.S.: Robust multi-object tracking to acquire object oriented videos in indoor sports. In: *Int. Conf. on Information and Communication Technology Convergence (ICTC)*, pp. 1104–1107. IEEE, 2016.

[137] Klaser A., Marszałek M., Schmid C.: A spatio-temporal descriptor based on 3D-gradients. In: *19th British Machine Vision Conf. (BMVC)*, pp. 275–1. British Machine Vision Association, 2008.

[138] Knudson D.V.: *Qualitative Diagnosis of Human Movement: Improving Performance in Sport and Exercise*. Human Kinetics, 2013.

[139] Kohavi R., et al.: A study of cross-validation and bootstrap for accuracy estimation and model selection. In: *Ijcai*, vol. 14, pp. 1137–1145. Montreal, Canada, 1995.

[140] Koppula H.S., Gupta R., Saxena A.: Learning human activities and object affordances from RGB-D videos. In: *The Int. J. of Robotics Research*, vol. 32(8), pp. 951–970, 2013.

[141] Krizhevsky A., Sutskever I., Hinton G.E.: Imagenet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems*, pp. 1097–1105. 2012.

[142] Krzeszowski T., Kwolek B., Michalczuk A., Świtoński A., Josiński H.: View independent human gait recognition using markerless 3D human motion capture. In: *Int. Conf. on Computer Vision and Graphics*, pp. 491–500. Springer, 2012.

[143] Kumada K., Usui Y., Kondo K.: Golf swing tracking and evaluation using Kinect sensor and particle filter. In: *Int. Symposium on Intelligent Signal Processing and Communications Systems (ISPACS)*, pp. 698–703. IEEE, 2013.

[144] Kwolek B.: Object tracking using grayscale appearance models and swarm based particle filter. In: *Int. Workshop on Hybrid Artificial Intelligence Systems*, pp. 433–440. Springer, 2008.

[145] Kwolek B., Kepski M.: Human fall detection on embedded platform using depth maps and wireless accelerometer. In: *Computer Methods and Programs in Biomedicine*, vol. 117(3), pp. 489–501, 2014.

[146] Kwolek B., Kepski M.: Improving fall detection by the use of depth sensor and accelerometer. In: *Neurocomputing*, vol. 168, pp. 637–645, 2015.

[147] Kwolek B., Kepski M.: Fuzzy inference-based fall detection using kinect and body-worn accelerometer. In: *Applied Soft Computing*, vol. 40, pp. 305–318, 2016.

[148] Kwolek B., Krzeszowski T., Michalczuk A., Josinski H.: 3D gait recognition using spatio-temporal motion descriptors. In: *Asian Conf. on Intelligent Information and Database Systems*, pp. 595–604. Springer, 2014.

[149] Laptev I.: On space-time interest points. In: *Int. J. of Computer Vision*, vol. 64(2-3), pp. 107–123, 2005.

[150] Laptev I., Marszalek M., Schmid C., Rozenfeld B.: Learning realistic human actions from movies. In: *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8. IEEE, 2008.

[151] Lara O.D., Labrador M.A.: A survey on human activity recognition using wearable sensors. In: *IEEE Communications Surveys and Tutorials*, vol. 15(3), pp. 1192–1209, 2013.

[152] Lei J., Ren X., Fox D.: Fine-grained kitchen activity recognition using RGB-D. In: *ACM Conf. on Ubiquitous Computing*, pp. 208–211. ACM, 2012.

[153] Li S., Pathirana P.N., Bonacci J.: A general pose estimation algorithm in a multi-Kinect system. In: *7th Int. Conf. on Information and Automation for Sustainability (ICIAFS)*, pp. 1–5. IEEE, 2014.

[154] Li W., Zhang Z., Liu Z.: Action recognition based on a bag of 3D points. In: *IEEE Conf. on Computer Vision and Pattern Recognition Workshops (CVPR Workhops)*, pp. 9–14. IEEE, 2010.

[155] Li Y., Wang Y., Huang W., Zhang Z.: Automatic image stitching using SIFT. In: *Int. Conf. on Audio, Language and Image Processing (ICALIP)*, pp. 568–571. IEEE, 2008.

[156] Liang B., Zheng L.: 3D motion trail model based pyramid histograms of oriented gradient for action recognition. In: *22nd Int. Conf. on Pattern Recognition (ICPR)*, pp. 1952–1957. IEEE, 2014.

[157] Liang B., Zheng L.: A survey on human action recognition using depth sensors. In: *Int. Conf. on Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1–8. IEEE, 2015.

[158] Liang D., Liu Y., Huang Q., Gao W.: A scheme for ball detection and tracking in broadcast soccer video. In: *Pacific-Rim Conf. on Multimedia*, pp. 864–875. Springer, 2005.

[159] Lin F., Chang C., Jou Y., Pan H., Hsu T.: The study of influence of fencing handle type and handle angle on wrist for a fencing game. In: *17th Int. Conf. on Industrial Engineering and Engineering Management (IE&EM)*, pp. 1624–1627. IEEE, 2010.

[160] Liu Z., Huang J., Han J., Bu S., Lv J.: Human motion tracking by multiple RGBD cameras. In: *IEEE Trans. on Circuits and Systems for Video Technology*, 2016.

[161] Lowe D.G.: Distinctive image features from scale-invariant keypoints. In: *Int. J. of Computer Vision*, vol. 60(2), pp. 91–110, 2004.

[162] Lu W.L., Little J.J.: Simultaneous tracking and action recognition using the PCA-HOG descriptor. In: *3rd Canadian Conf. on Computer and Robot Vision*, pp. 6–6. IEEE, 2006.

[163] Lv F., Nevatia R.: Single view human action recognition using key pose matching and viterbi path searching. In: *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8. IEEE, 2007.

[164] Macknojia R., Chávez-Aragón A., Payeur P., Laganiere R.: Calibration of a network of Kinect sensors for robotic inspection over a large workspace. In: *IEEE Workshop on Robot Vision (WORV)*, pp. 184–190. IEEE, 2013.

[165] Madgwick S.O., Harrison A.J., Vaidyanathan R.: Estimation of IMU and MARG orientation using a gradient descent algorithm. In: *IEEE Int. Conf. on Rehabilitation Robotics (ICORR)*, pp. 1–7. IEEE, 2011.

[166] Malawski F.: Fencing Footwork Dataset (FFD). `http://home.agh.edu.pl/~fmal/ffd/`. Last access on Jan 2019.

[167] Malawski F.: Applying hand gesture recognition with time-of-flight camera for 3D medical data analysis. In: *Challenges of Modern Technology*, vol. 5, 2014.

[168] Malawski F.: Top-view people counting in public transportation using Kinect. In: *Challenges of Modern Technology*, vol. 5, 2014.

[169] Malawski F.: Acquisition of databases for facial analysis. In: *Challenges of Modern Technology*, vol. 7(3), pp. 3–7, 2016.

[170] Malawski F.: Driver assistance system using augmented reality headset. In: *41st IEEE Int. Conf. on Telecommunications and Signal Processing (TSP)*, pp. 1–4. IEEE, 2018.

[171] Malawski F.: Real-time first person perspective tracking and feedback system for weapon practice support in fencing. In: *Applications of Intelligent Systems (APPIS)*, vol. 310, pp. 79 – 88, 2018.

[172] Malawski F., Gałka J.: Framework for automated customer service in sign language. In: *24th Conf. on Computer Graphics, Visualization and Computer Vision (WSCG)*. Václav Skala - UNION Agency, 2016.

[173] Malawski F., Gałka J.: System for multimodal data acquisition for human action recognition. In: *Multimedia Tools and Applications*, vol. 7(18), pp. 1–26, Springer, 2018.

[174] Malawski F., Kwolek B.: Classification of basic footwork in fencing using accelerometer. In: *Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*, vol. 6, p. 20. 2016.

[175] Malawski F., Kwolek B.: Real-time action detection and analysis in fencing footwork. In: *40th IEEE Int. Conf. on Telecommunications and Signal Processing (TSP)*, pp. 520–523. IEEE, 2017.

[176] Malawski F., Kwolek B.: Improving multimodal action representation with joint motion history context. In: *J. of Visual Communication and Image Representation*, vol. 61, pp. 198–208, 2019.

[177] Malawski F., Kwolek B.: Recognition of action dynamics in fencing using multimodal cues. In: *Image and Vision Computing*, vol. 75, pp. 1–10, Elsevier, 2018.

[178] Malawski F., Kwolek B., Sako S.: Using Kinect for facial expression recognition under varying poses and illumination. In: *Int. Conf. on Active Media Technology (AMT)*, pp. 395–406. Springer, 2014.

[179] Manafifard M., Ebadi H., Moghaddam H.A.: A survey on player tracking in soccer videos. In: *Computer Vision and Image Understanding*, 2017.

[180] Mandel J.: *The statistical analysis of experimental data.* Courier Corporation, 2012.

[181] Mangai U.G., Samanta S., Das S., Chowdhury P.R.: A survey of decision fusion and feature fusion strategies for pattern classification. In: *IETE Technical Review*, vol. 27(4), pp. 293–307, 2010.

[182] Mantovani G., Ravaschio A., Piaggi P., Landi A.: Fine classification of complex motion pattern in fencing. In: *Procedia Engineering*, vol. 2(2), pp. 3423–3428, 2010.

[183] Maqueda A.I., del Blanco C.R., Jaureguizar F., García N.: Human-computer interaction based on visual hand-gesture recognition using volumetric spatiograms of local binary patterns. In: *Computer Vision and Image Understanding*, vol. 141, pp. 126–137, 2015.

[184] Margarito J., Helaoui R., Bianchi A.M., Sartor F., Bonomi A.G.: User-independent recognition of sports activities from a single wrist-worn accelerometer: A template-matching-based approach. In: *IEEE Trans. on Biomedical Engineering*, vol. 63(4), pp. 788–796, 2016.

[185] Marin G., Dominio F., Zanuttigh P.: Hand gesture recognition with jointly calibrated leap motion and depth sensor. In: *Multimedia Tools and Applications*, vol. 75(22), pp. 14991–15015, 2016.

[186] Matikainen P., Hebert M., Sukthankar R.: Trajectons: Action recognition through the motion analysis of tracked features. In: *12th IEEE Int. Conf. on Computer Vision Workshops (ICCV Workshops)*, pp. 514–521. IEEE, 2009.

[187] Maurer U., Smailagic A., Siewiorek D.P., Deisher M.: Activity recognition and monitoring using multiple sensors on different body positions. In: *Int. Workshop on Wearable and Implantable Body Sensor Networks (BSN)*, pp. 4–pp. IEEE, 2006.

[188] Mauthner T., Koch C., Tilp M., Bischof H.: Visual tracking of athletes in beach volleyball using a single camera. In: *Int. J. of Computer Science in Sport*, vol. 6(2), pp. 21–34, 2007.

[189] McCall C., Reddy K.K., Shah M.: Macro-class selection for hierarchical k-NN classification of inertial sensor data. In: *2nd Int. Conf. on Pervasive and Embedded Computing and Communication Systems (PECCS)*, pp. 106–114. 2012.

[190] Messer K., Christmas W., Kittler J.: Automatic sports classification. In: *16th Int. Conf. on Pattern Recognition (ICPR)*, vol. 2, pp. 1005–1008. IEEE, 2002.

[191] Messing R., Pal C., Kautz H.: Activity recognition using the velocity histories of tracked keypoints. In: *12th IEEE Int. Conf. on Computer Vision (ICCV)*, pp. 104–111. IEEE, 2009.

[192] Microsoft: Kinect SDK skeletal tracking. `https://msdn.microsoft.com/pl-pl/library/kinect-sdk--skeletal-tracking.aspx`. Last access on Jan 2019.

[193] Moeslund T.B., Hilton A., Krüger V.: A survey of advances in vision-based human motion capture and analysis. In: *Computer Vision and Image Understanding*, vol. 104(2), pp. 90–126, 2006.

[194] Moore K.C., Chow F.M., Chow J.Y.: Novel lunge biomechanics in modern sabre fencing. In: *Procedia Engineering*, vol. 112, pp. 473–478, 2015.

[195] Morales J., Akopian D.: Physical activity recognition by smartphones, a survey. In: *Biocybernetics and Biomedical Engineering*, vol. 37(3), pp. 388–400, 2017.

[196] Mori G., Malik J.: Recovering 3D human body configurations using shape contexts. In: *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 28(7), pp. 1052–1062, 2006.

[197] Mori G., Ren X., Efros A.A., Malik J.: Recovering human body configurations: Combining segmentation and recognition. In: *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. II–II. IEEE, 2004.

[198] Morimitsu H., Bloch I., Cesar-Jr R.M.: Exploring structure for long-term tracking of multiple objects in sports videos. In: *Computer Vision and Image Understanding*, 2016.

[199] Morris J.: Accelerometry - a technique for the measurement of human body movements. In: *J. of Biomechanics*, vol. 6(6), pp. 729–736, 1973.

[200] Müller M., Röder T., Clausen M., Eberhardt B., Krüger B., Weber A.: Documentation mocap database HDM05. Tech. Rep. CG-2007-2, Universität Bonn, 2007.

[201] Negin F., Özdemir F., Akgül C.B., Yüksel K.A., Erçil A.: A decision forest based feature selection framework for action recognition from RGB-depth cameras. In: *Int. Conf. Image Analysis and Recognition (ICIAR)*, pp. 648–657. Springer, 2013.

[202] Ngo T.T., Makihara Y., Nagahara H., Mukaigawa Y., Yagi Y.: Similar gait action recognition using an inertial sensor. In: *Pattern Recognition*, vol. 48(4), pp. 1289–1301, 2015.

[203] Niebles J.C., Chen C.W., Fei-Fei L.: Modeling temporal structure of decomposable motion segments for activity classification. In: *European Conf. on Computer Vision (ECCV)*, pp. 392–405. Springer, 2010.

[204] Nielsen J.: *Usability Engineering*. Academic Press, 1994.

[205] NIST: Metric in sports. `https://www.nist.gov/pml/weights-and-measures/metric-sports`. Last access on Jan 2019.

[206] Nyan M., Tay F., Seah K., Sitoh Y.: Classification of gait patterns in the time-frequency domain. In: *J. of Biomechanics*, vol. 39(14), pp. 2647–2656, 2006.

[207] O'Donovan K.J., Kamnik R., O'Keeffe D.T., Lyons G.M.: An inertial and magnetic sensor based technique for joint angle measurement. In: *J. of Biomechanics*, vol. 40(12), pp. 2604–2611, 2007.

[208] Ojala T., Pietikainen M., Maenpaa T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. In: *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24(7), pp. 971–987, 2002.

[209] OpenGL ES. `https://www.khronos.org/opengles/`. Last access on Jan 2019.

[210] Oreifej O., Liu Z.: HON4D: Histogram of oriented 4D normals for activity recognition from depth sequences. In: *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 716–723. 2013.

[211] Tejero-de Pablos A., Nakashima Y., Sato T., Yokoya N.: Human action recognition-based video summarization for RGB-D personal sports video. In: *IEEE Int. Conf. on Multimedia and Expo (ICME)*, pp. 1–6. IEEE, 2016.

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

[213] Parisot P., De Vleeschouwer C.: Scene-specific classifier for effective and efficient team sport players detection from a single calibrated camera. In: *Computer Vision and Image Understanding*, 2017.

[214] Parkka J., Ermes M., Korpipaa P., Mantyjarvi J., Peltola J., Korhonen I.: Activity classification using realistic data from wearable sensors. In: *IEEE Trans. on Information Technology in Biomedicine*, vol. 10(1), pp. 119–128, 2006.

[215] PBT Fencing. `http://www.pbtfencing.com/`. Last access on Jan 2019.

[216] Pham H.H., Khoudour L., Crouzil A., Zegers P., Velastin S.A.: Exploiting deep residual networks for human action recognition from skeletal data. In: *Computer Vision and Image Understanding*, 2018.

[217] PhaseSpace: Impulse X2 Motion Capture System. `http://www.phasespace.com/impulse-motion-capture.html`. Last access on Jan 2019.

[218] Phulkar A.: Tactics, Techniques & Skills. `https://www.slideshare.net/AshishPhulkar/tactics-technique-and-skills-training`. Last access on Jan 2019.

[219] Piccardi M.: Background subtraction techniques: A review. In: *IEEE Int. Conf. on Systems, Man and Cybernetics*, vol. 4, pp. 3099–3104. IEEE, 2004.

[220] Picerno P.: 25 years of lower limb joint kinematics by using inertial and magnetic sensors: A review of methodological approaches. In: *Gait & Posture*, vol. 51, pp. 239–246, 2017.

[221] Platt J.: Fast training of support vector machines using sequential minimal optimization. In: B. Schoelkopf, C. Burges, A. Smola, eds., *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1998.

[222] Poppe R.: Vision-based human motion analysis: An overview. In: *Computer Vision and Image Understanding*, vol. 108(1), pp. 4–18, 2007.

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

[224] Powers D.M.: Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. In: *J. of Machine Learning Technologies*, vol. 2(1), pp. 37–63, 2011.

[225] Presti L.L., La Cascia M.: 3D skeleton-based human action classification: A survey. In: *Pattern Recognition*, vol. 53, pp. 130–147, 2016.

[226] Reily B., Zhang H., Hoff W.: Real-time gymnast detection and performance analysis with a portable 3D camera. In: *Computer Vision and Image Understanding*, 2016.

[227] Reno V., Mosca N., Nitti M., D'Orazio T., Guaragnella C., Campagnoli D., Prati A., Stella E.: A technology platform for automatic high-level tennis game analysis. In: *Computer Vision and Image Understanding*, 2017.

[228] Rice S.G., et al.: Medical conditions affecting sports participation. In: *Pediatrics*, vol. 121(4), pp. 841–848, 2008.

[229] Rodriguez M.: Spatio-temporal maximum average correlation height templates in action recognition and video summarization. In: *Electronic Theses and Dissertations*, 4323. University of Central Florida, 2010.

[230] Roetenberg D., Luinge H.J., Baten C.T., Veltink P.H.: Compensation of magnetic disturbances improves inertial and magnetic sensing of human body segment orientation. In: *IEEE Trans. on Neural Systems and Rehabilitation Engineering*, vol. 13(3), pp. 395–405, 2005.

[231] Rumelhart D.E., Hinton G.E., Williams R.J.: Learning internal representations by error propagation. Tech. rep., University of California San Diego Institute for Cognitive Science, 1985.

[232] RunnersWorld: Heart rate monitors - the basics. `https://www.runnersworld.com/uk/gear/a760496/heart-rate-monitors-the-basics`. Last access on Jan 2019.

[233] Sabatini A.M.: Quaternion-based extended Kalman filter for determining orientation by inertial and magnetic sensing. In: *IEEE Trans. on Biomedical Engineering*, vol. 53(7), pp. 1346–1356, 2006.

[234] Safdarnejad S.M., Liu X., Udpa L., Andrus B., Wood J., Craven D.: Sports videos in the wild (SVW): A video dataset for sports analysis. In: *11th IEEE Int. Conf. and Workshops on Automatic Face and Gesture Recognition (FG)*, vol. 1, pp. 1–7. IEEE, 2015.

[235] Sagawa K., Ohkubo K.: 2D trajectory estimation during free walking using a tiptoe-mounted inertial sensor. In: *J. of Biomechanics*, vol. 48(10), pp. 2054–2059, 2015.

[236] Sarbolandi H., Lefloch D., Kolb A.: Kinect range sensing: Structured-light versus time-of-flight Kinect. In: *Computer Vision and Image Understanding*, vol. 139, pp. 1–20, 2015.

[237] Schrefer S., Como D., Myers T.: *Mosby's Medical, Nursing, & Allied Health Dictionary*. Philadelphia, Mosby, 2002.

[238] Schuldt C., Laptev I., Caputo B.: Recognizing human actions: a local SVM approach. In: *17th Int. Conf. on Pattern Recognition (ICPR)*, vol. 3, pp. 32–36. IEEE, 2004.

[239] Scovanner P., Ali S., Shah M.: A 3-dimensional SIFT descriptor and its application to action recognition. In: *15th ACM Int. Conf. on Multimedia*, pp. 357–360. ACM, 2007.

[240] Se S., Lowe D., Little J.: Vision-based mobile robot localization and mapping using scale-invariant features. In: *IEEE Int. Conf. on Robotics and Automation (ICRA)*, vol. 2, pp. 2051–2058. IEEE, 2001.

[241] Seddik B., Gazzah S., Amara N.E.B.: Human action recognition using a multi-layered fusion scheme of Kinect modalities. In: *IET Computer Vision*, 2017.

[242] Setti F., Conigliaro D., Rota P., Bassetti C., Conci N., Sebe N., Cristani M.: The S-Hock dataset: A new benchmark for spectator crowd analysis. In: *Computer Vision and Image Understanding*, 2017.

[243] Sharma A., Agarwal M., Sharma A., Dhuria P.: Motion capture process, techniques and applications. In: *Int. J. on Recent and Innovation Trends in Computing and Communication*, vol. 1(4), pp. 251–257, 2013.

[244] Shotton J., Girshick R., Fitzgibbon A., Sharp T., Cook M., Finocchio M., Moore R., Kohli P., Criminisi A., Kipman A., et al.: Efficient human pose estimation from single depth images. In: *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 35(12), pp. 2821–2840, 2013.

[245] Simonyan K., Zisserman A.: Two-stream convolutional networks for action recognition in videos. In: *Advances in Neural Information Processing Systems*, pp. 568–576. 2014.

[246] Sivic J., Zisserman A.: Efficient visual search of videos cast as text retrieval. In: *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31(4), pp. 591–606, 2009.

[247] Spriggs E.H., De La Torre F., Hebert M.: Temporal segmentation and activity classification from first-person sensing. In: *IEEE Conf. On Computer Vision and Pattern Recognition Workshops (CVPR Workshops)*, pp. 17–24. IEEE, 2009.

[248] Srivastava N., Mansimov E., Salakhudinov R.: Unsupervised learning of video representations using lstms. In: *Int. Conf. on Machine Learning (ICML)*, pp. 843–852. 2015.

[249] Stallings L.M.: *Motor Learning: From Theory to Practice*. Mosby Inc., 1982.

[250] Stąpor K.: *Metody Klasyfikacji Obiektów w Wizji Komputerowej*. Wydawnictwo Naukowe PWN, 2011.

[251] Stąpor K.: Evaluation of classifiers: current methods and future research directions. In: *Federated Conf. on Computer Science and Information Systems (FedCSIS)*, pp. 37–40. 2017.

[252] Strava. `https://www.strava.com`. Last access on Jan 2019.

[253] Sutskever I., Martens J., Dahl G., Hinton G.: On the importance of initialization and momentum in deep learning. In: *Int. Conf. on Machine Learning (ICML)*, pp. 1139–1147. 2013.

[254] Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., Erhan D., Vanhoucke V., Rabinovich A.: Going deeper with convolutions. In: *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9. 2015.

[255] Technologies X.I.: x-IMU sensor. `http://x-io.co.uk/x-imu/`. Last access on Jan 2019.

[256] Teu K.K., Kim W., Fuss F.K., Tan J.: The analysis of golf swing as a kinematic chain using dual Euler angle algorithm. In: *J. of Biomechanics*, vol. 39(7), pp. 1227–1238, 2006.

[257] Thomas G., Gade R., Moeslund T.B., Carr P., Hilton A.: Computer vision for sports: Current applications and research topics. In: *Computer Vision and Image Understanding*, 2017.

[258] Thurau C., Hlavác V.: Pose primitive based human action recognition in videos or still images. In: *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8. IEEE, 2008.

[259] Tian X., Fan J.: Joints kinetic and relational features for action recognition. In: *Signal Processing*, vol. 142, pp. 412–422, 2018.

[260] Tibshirani R.: Regression shrinkage and selection via the lasso. In: *J. of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[261] Tran D., Bourdev L., Fergus R., Torresani L., Paluri M.: Learning spatiotemporal features with 3d convolutional networks. In: *IEEE Int. Conf. on Computer Vision (ICCV)*, pp. 4489–4497. 2015.

[262] Tran D., Sorokin A.: Human activity recognition with metric learning. In: *European Conf. on Computer Vision (ECCV)*, pp. 548–561. Springer, 2008.

[263] Turaga P., Chellappa R., Subrahmanian V.S., Udrea O.: Machine recognition of human activities: A survey. In: *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 18(11), pp. 1473–1488, 2008.

[264] Turchini F., Seidenari L., Del Bimbo A.: Understanding and localizing activities from correspondences of clustered trajectories. In: *Computer Vision and Image Understanding*, 2016.

[265] Ullah A., Ahmad J., Muhammad K., Sajjad M., Baik S.W.: Action recognition in video sequences using deep Bi-directional LSTM with CNN features. In: *IEEE Access*, vol. 6, pp. 1155–1166, 2018.

[266] Urtasun R., Fleet D.J., Fua P.: Monocular 3D tracking of the golf swing. In: *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. 932–938. IEEE, 2005.

[267] Van Krevelen D., Poelman R.: A survey of augmented reality technologies, applications and limitations. In: *Int. J. of Virtual Reality*, vol. 9(2), p. 1, 2010.

[268] Vieira A.W., Nascimento E.R., Oliveira G.L., Liu Z., Campos M.F.: Stop: Space-time occupancy patterns for 3D action recognition from depth map sequences. In: *Iberoamerican Congress on Pattern Recognition*, pp. 252–259. Springer, 2012.

[269] Vieira A.W., Nascimento E.R., Oliveira G.L., Liu Z., Campos M.F.: On the improvement of human action recognition from depth map sequences using Space-Time Occupancy Patterns. In: *Pattern Recognition Letters*, vol. 36, pp. 221–227, 2014.

[270] Viola P., Jones M.J.: Robust real-time face detection. In: *Int. J. of Computer Vision*, vol. 57(2), pp. 137–154, 2004.

[271] Wang H., Kläser A., Schmid C., Liu C.L.: Action recognition by dense trajectories. In: *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 3169–3176. IEEE, 2011.

[272] Wang H., Schmid C.: Action recognition with improved trajectories. In: *IEEE Int. Conf. on Computer Vision (ICCV)*, pp. 3551–3558. 2013.

[273] Wang H., Ullah M.M., Klaser A., Laptev I., Schmid C.: Evaluation of local spatio-temporal features for action recognition. In: *British Machine Vision Conf. (BMVC)*, pp. 124–1. BMVA Press, 2009.

[274] Wang H., Wang L.: Beyond joints: Learning representations from primitive geometries for skeleton-based action recognition and detection. In: *IEEE Trans. on Image Processing*, vol. 27(9), pp. 4382–4394, 2018.

[275] Wang J., Liu Z., Chorowski J., Chen Z., Wu Y.: Robust 3D action recognition with random occupancy patterns. In: *European Conf. on Computer Vision (ECCV)*, pp. 872–885. Springer, 2012.

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

[277] Wang L., Qiao Y., Tang X.: Action recognition with trajectory-pooled deep-convolutional descriptors. In: *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 4305–4314. 2015.

[278] Wang L., Suter D.: Informative shape representations for human action recognition. In: *18th Int. Conf. on Pattern Recognition (ICPR)*, vol. 2, pp. 1266–1269. IEEE, 2006.

[279] Wang P., Li W., Gao Z., Tang C., Ogunbona P.O.: Depth pooling based large-scale 3-D action recognition with convolutional neural networks. In: *IEEE Trans. on Multimedia*, vol. 20(5), pp. 1051–1061, 2018.

[280] Wang P., Li W., Li C., Hou Y.: Action recognition based on joint trajectory maps with convolutional neural networks. In: *Knowledge-Based Systems*, 2018.

[281] Wang P., Li W., Ogunbona P., Wan J., Escalera S.: RGB-D-based human motion recognition with deep learning: A survey. In: *Computer Vision and Image Understanding*, 2018.

[282] Wang X., Ablavsky V., Shitrit H.B., Fua P.: Take your eyes off the ball: Improving ball-tracking by focusing on team play. In: *Computer Vision and Image Understanding*, vol. 119, pp. 102–115, 2014.

[283] Wang X., Farhadi A., Gupta A.: Actions˜transformations. In: *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 2658–2667. 2016.

[284] Warburton D.E., Nicol C.W., Bredin S.S.: Health benefits of physical activity: The evidence. In: *Canadian Medical Association J.*, vol. 174(6), pp. 801–809, 2006.

[285] Weinland D., Özuysal M., Fua P.: Making action recognition robust to occlusions and viewpoint changes. In: *European Conf. on Computer Vision (ECCV)*, pp. 635–648. Springer, 2010.

[286] Weinland D., Ronfard R., Boyer E.: Motion history volumes for free viewpoint action recognition. In: *Workshop on Modeling People and Human Interaction (PHI)*. 2005.

[287] Weinland D., Ronfard R., Boyer E.: A survey of vision-based methods for action representation, segmentation and recognition. In: *Computer Vision and Image Understanding*, vol. 115(2), pp. 224–241, 2011.

[288] Weiss G.M., Timko J.L., Gallagher C.M., Yoneda K., Schreiber A.J.: Smartwatch-based activity recognition: A machine learning approach. In: *IEEE-EMBS Int. Conf. on Biomedical and Health Informatics (BHI)*, pp. 426–429. IEEE, 2016.

[289] Weka Data Mining Software. `https://www.cs.waikato.ac.nz/ml/weka/`. Last access on Jan 2019.

[290] Wikipedia: Fencing foil valid surfaces. `https://commons.wikimedia.org/wiki/File:Fencing_foil_valid_surfaces_2009.svg/`. Last access on Jan 2019.

[291] Wikipedia: Small sword. `https://en.wikipedia.org/wiki/Small_sword`. Last access on Jan 2019.

[292] Wikipedia: The lines in fencing. `https://commons.wikimedia.org/wiki/File:The_lines_in_Fencing.png?uselang=en`. Last access on Jan 2019.

[293] Willems G., Tuytelaars T., Van Gool L.: An efficient dense and scale-invariant spatio-temporal interest point detector. In: *European Conf. on Computer Vision (ECCV)*, pp. 650–663, 2008.

[294] Williams L., Walmsley A.: Response timing and muscular coordination in fencing: A comparison of elite and novice fencers. In: *J. of Science and Medicine in Sport*, vol. 3(4), pp. 460–475, 2000.

[295] Windridge D., Kittler J., De Campos T., Yan F., Christmas W., Khan A.: A novel Markov logic rule induction strategy for characterizing sports video footage. In: *IEEE Multimedia*, vol. 22(2), pp. 24–35, 2015.

[296] Worsey M.T., Espinosa H.G., Shepherd J.B., Thiel D.V.: Inertial sensors for performance analysis in combat sports: A systematic review. In: *Sports*, vol. 7(1), p. 28, 2019.

[297] Wu J., Sun L., Jafari R.: A wearable system for recognizing American sign language in real-time using IMU and surface EMG sensors. In: *IEEE J. of Biomedical and Health Informatics*, vol. 20(5), pp. 1281–1290, 2016.

[298] Xia L., Chen C.C., Aggarwal J.: View invariant human action recognition using histograms of 3D joints. In: *IEEE Conf. on Computer Vision and Pattern Recognition Workshops (CVPR Workshops)*, pp. 20–27. IEEE, 2012.

[299] Yang X., Tian Y.: Effective 3D action recognition using EigenJoints. In: *J. of Visual Communication and Image Representation*, vol. 25(1), pp. 2–11, 2014.

[300] Ye M., Zhang Q., Wang L., Zhu J., Yang R., Gall J.: A survey on human motion analysis from depth data. In: *Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications*, pp. 149–187. Springer, 2013.

[301] Yovcheva Z., Buhalis D., Gatzidis C., van Elzakker C.P.: Empirical evaluation of smartphone augmented reality browsers in an urban tourism destination context. In: *Int. J. of Mobile Human Computer Interaction (IJMHCI)*, vol. 6(2), pp. 10–31, 2014.

[302] Yue-Hei Ng J., Hausknecht M., Vijayanarasimhan S., Vinyals O., Monga R., Toderici G.: Beyond short snippets: Deep networks for video classification. In: *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 4694–4702. 2015.

[303] Yun K., Honorio J., Chattopadhyay D., Berg T.L., Samaras D.: Two-person interaction detection using body-pose features and multiple instance learning. In: *IEEE Conf. on Computer Vision and Pattern Recognition Workshops (CVPR Workshops)*, pp. 28–35. IEEE, 2012.

[304] Zeng R., Lakemond R., Denman S., Sridharan S., Fookes C., Morgan S.: Vertical axis detection for sport video analytics. In: *Int. Conf. on Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1–7. IEEE, 2016.

[305] Zhang C., Tian Y., Guo X., Liu J.: DAAL: Deep activation-based attribute learning for action recognition in depth videos. In: *Computer Vision and Image Understanding*, vol. 167, pp. 37–49, 2018.

[306] Zhang C., Yang F., Li G., Zhai Q., Jiang Y., Xuan D.: MV-Sports: A motion and vision sensor integration-based sports analysis system. In: *IEEE Conf. on Computer Communications (INFOCOM)*, pp. 1070–1078. IEEE, 2018.

[307] Zhang H., Parker L.E.: 4-dimensional local spatio-temporal features for human activity recognition. In: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pp. 2044–2049. IEEE, 2011.

[308] Zhang H., Zhong P., He J., Xia C.: Combining depth-skeleton feature with sparse coding for action recognition. In: *Neurocomputing*, vol. 230, pp. 417–426, 2017.

[309] Zhang J., Li W., Ogunbona P.O., Wang P., Tang C.: RGB-D-based action recognition datasets: A survey. In: *Pattern Recognition*, vol. 60, pp. 86–105, 2016.

[310] Zhang L., Hsieh J.C., Ting T.T., Huang Y.C., Ho Y.C., Ku L.K.: A Kinect based golf swing score and grade system using GMM and SVM. In: *5th Int. Congress on Image and Signal Processing (CISP)*, pp. 711–715. IEEE, 2012.

[311] Zhang Z., Hu Y., Chan S., Chia L.T.: Motion context: A new representation for human action recognition. In: *European Conf. on Computer Vision (ECCV)*, pp. 817–829, 2008.

[312] Zhu F., Shao L., Xie J., Fang Y.: From handcrafted to learned representations for human action recognition: A survey. In: *Image and Vision Computing*, vol. 55, pp. 42–52, 2016.

[313] Zhu G., Huang Q., Xu C., Rui Y., Jiang S., Gao W., Yao H.: Trajectory based event tactics analysis in broadcast sports video. In: *15th ACM Int. Conf. on Multimedia*, pp. 58–67. ACM, 2007.

[314] Zhu R., Zhou Z.: A real-time articulated human motion tracking using tri-axis inertial/magnetic sensors package. In: *IEEE Trans. on Neural Systems and Rehabilitation Engineering*, vol. 12(2), pp. 295–302, 2004.